GEIDEA IOS APP APPLICATION ASSESSMENT

draft

Contents

[Organization and communication 3](#_Toc93484231)

[Missing scalability 4](#_Toc93484232)

[TECHNICAL 4](#_Toc93484233)

[action points 6](#_Toc93484234)

[cleanup 6](#_Toc93484235)

# Organization and communication

* Not a real scrum is used, it is like a waterfall approach. No estimations on complexity level features only timeframes.
* Stories and tasks not tracked in written form (e.g. JIRA issues)
* Amending requirements of a story in spoken not written form
* We do not participate in estimations but expected to meet deadlines
* Change of requirements near the completion of a story/feature or even long after it is formally closed
* We do not have access to the main git repository
* no access to Confluence and JIRA, stories sent as PDFs
* Code not reviewed/merged for weeks even months after has been submitted.
* All features like QR Payment, EInvoice, Onboarding etc. Are still in pending state because of missing scalability features and Geidea team development involvement in iOS
* Very hard to merge on Geidea repo git because we worked for almost a year separate in a cloned repository. Git conflicts will be extremely hard to solve
* I was not added to Apple development team, so I can not use the real Iphone device. Only simulator can be used, but now there are 2 separate architectures based on hardware, Macbook with Apple chip M1 or Macbook with Intel chip. I needed to make some changes on the project setup to fix this issue. I cannot build the app and distribute for testing.

# Missing scalability

* Very old master branch from March 2021
* There is no Country Selection, app is working only on Saudi
* The currency is hardcoded as SAR in every request
* The app connects to PROD and Preprod environment, changing the environment by comment and uncomment lines of code
* One class of constants with all the values 675 lines of code
* Sync class (the network layer) 5347 lines of code. Conflicts hard to fix
* Only huge storyboard. Conflicts are impossible to fix. No team or collaboration oriented programming
* There is no single Beckend for the App
* App connects directly to several service provider. (This must be done by a single beckend connection): E.G.
* EInvoiceServer
* EPOS
* KeyCloack
* GSDKServer
* EReceipt
* OnBoarding (MMS)
* PGW using GeideaPaymentSDK

# TECHNICAL

* MVC architecture without any pattern applied. All business logic and UI in same place, making development hard to follow, hard to maintain, hard to test
* No presentation architectural pattern used (MVP, MVVM, etc.)
* Bad networking layer, headers are hardcoded for every request, without any design patten applied (5347 lines of code) impossible to follow
* Bad data refreshing mechanism. E.G. Transactions can be refreshed by killing the app or switch tab
* Almost no separation of concerns
* Duplication of networking layer we have Keycloak and GSDK with the same endpoints, and the requests for gsdk uses Keycloak enpoints. Needed to change this for making adding Egypt stuff
* No models for the networking layer
* No Parsing layer, all fields are hardcoded when needed
* There are no components that can be reused in networking e.g. models, headers, completions
* Some classes are huge ex. PaymentMethodsViewController have 1902 lines of code because all logic needed is done in one place
* The folder structure is good, spitted by features and reusable UI components, But there are some old classes and folders not used that can be deleted and maintained by git repository

# action points

## cleanup

* Remove all unused libraries, modules, classes, methods, resources
* Remove legacy code inside existing
* Update all dependencies to latest
* Separate hardware-specific in reusable components
* Make Environment specific change into an ENUM that support scalability to other countries and Development easy
* Delegete responsabilities for the logic in separate classes that can be reused, maintained and Unit tested.
* All fields in parsing network are harcoded. Should be estracted in models for maintenance and sacalability