<CLIENT> <PROJECT> Warehouse Analytics: <PROJECT> Now

Mobile Application

Solution Overview

Nagarro Software Pvt. Ltd.

COE Mobility

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| Version | Date | Author/Contributor | Comments |
| 0.2 | 4-1-2018 | Kumar Bibek, Vishal Dwivedi | Document Created |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1 Our understanding of existing system 4](#_Toc502845798)

[2 Goals 4](#_Toc502845799)

[2.1 Business goals 4](#_Toc502845800)

[2.2 Design goals 5](#_Toc502845801)

[3 Non-functional requirements 5](#_Toc502845802)

[4 Solution overview 7](#_Toc502845803)

[4.1 Key design considerations 7](#_Toc502845804)

[4.1.1 Platform independence 7](#_Toc502845805)

[4.1.2 Reliability 7](#_Toc502845806)

[4.1.3 Security 7](#_Toc502845807)

[4.2 Solution Concept 8](#_Toc502845808)

[4.3 Mobile Application 8](#_Toc502845809)

[4.3.1 Logical architecture 8](#_Toc502845810)

[4.3.1.1 Mode-View-ViewModel 9](#_Toc502845811)

[4.3.1.2 Object oriented design 9](#_Toc502845812)

[4.3.1.3 Cross cutting concerns 10](#_Toc502845813)

[4.3.1.4 Solution for key functionalities 10](#_Toc502845814)

[4.3.1.4.1 Offline mode and Auto-synchronization of data 10](#_Toc502845815)

[4.3.1.5 Dashboard and Analytics Reports 12](#_Toc502845816)

[4.3.1.6 Forms 12](#_Toc502845817)

[4.3.1.7 Security 12](#_Toc502845818)

[4.3.1.7.1 Authentication 12](#_Toc502845819)

[4.3.1.7.2 Encryption 13](#_Toc502845820)

[4.3.1.8 Integrations 13](#_Toc502845821)

[4.3.1.8.1 Secure VPN Connection 13](#_Toc502845822)

[5 Technology 14](#_Toc502845823)

[5.1 Mobile Application 14](#_Toc502845824)

[6 Assumptions 14](#_Toc502845825)

[7 Scope considerations 16](#_Toc502845826)

[7.1 In-scope 16](#_Toc502845827)

[7.2 Out scope 16](#_Toc502845828)

# Our understanding of existing system

The (<PROJECT>) has initiated the development of a new analytics platform which will allow its users to more easily access, analyze, and visualize existing data resources to deliver useful insights for their operations and facilitate <PROJECT> in planning and optimizing its functions.

At present, <PROJECT> uses legacy mainframe system (BIS) that is not optimized for analytics reporting. Also, the current system is fragmented and inconsistent; data is available across multiple systems that needs to be cleansed and normalized. Thus, <PROJECT> needs to manually research the information to start each new analytics project, which can cause unnecessary delays.

Below are the key challenges-

1. Paper forms are used to collect information by the field inspectors which are then manually tabulated by a <PROJECT> personnel
2. Overall process is time consuming due to manual activity involved and process inefficiencies
3. Inspectors are required to often visit sites, carry multiple inspection forms and manually enter the details in the paper forms
4. Inspectors do not have access to any information on the field
5. In emergency situations, the process of data collection and processing takes multiple days which is not desired

The new analytics platform and the accompanying mobile application will assist the <PROJECT> and the inspectors on the field in eliminating a lot of manual paper work and making the process of collecting information quicker and more reliable.

# Goals

The primary business requirement for the mobile application is to assist the <PROJECT> staff for efficiently collecting data and viewing reports.

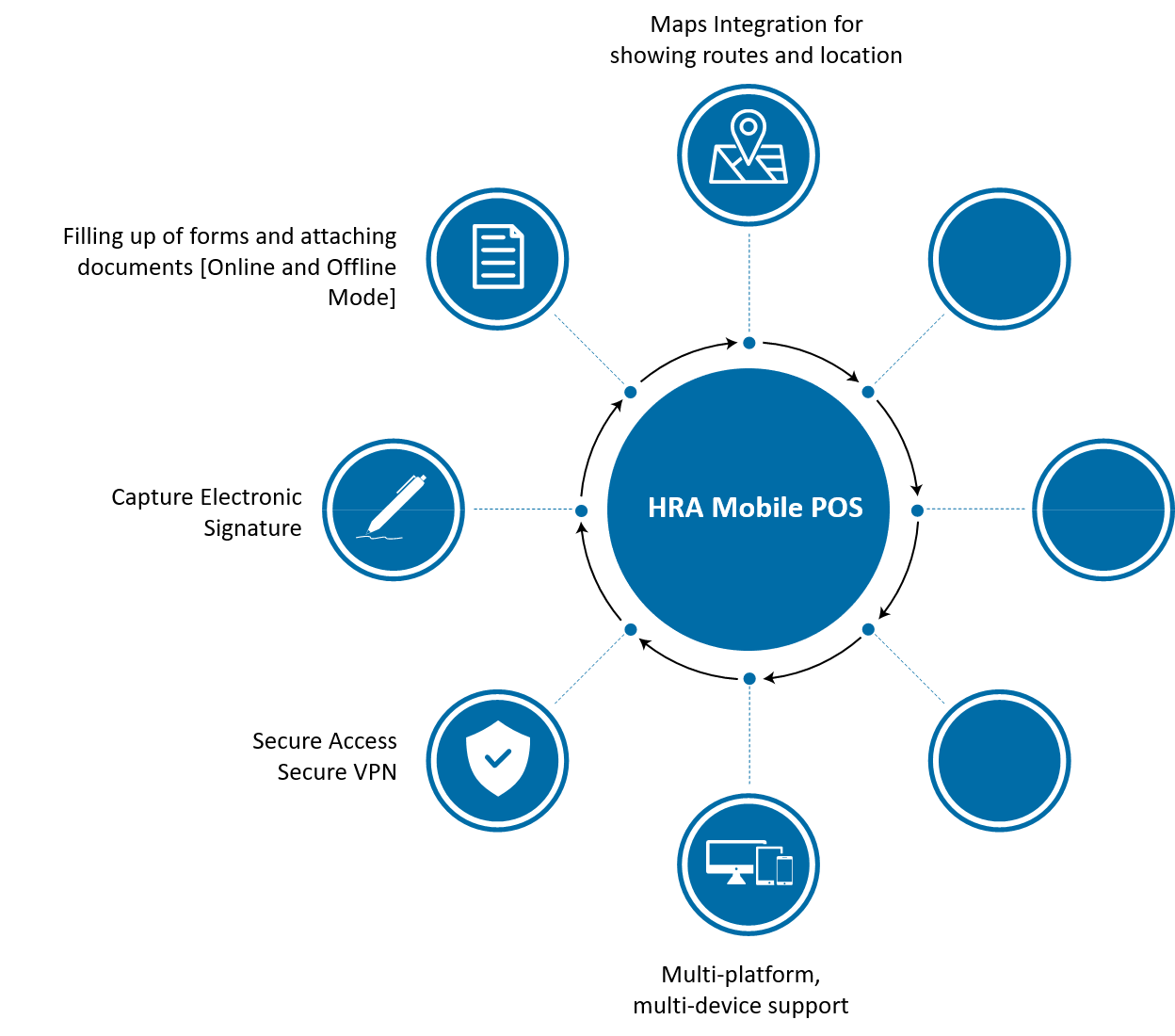
Key goals of the new system will be-

* Communication of allocations to inspectors through channels like push notifications, SMS, and email
* Collection of inspection data directly through a mobile application
* Visualization of analytics reports

## Business goals

Based on the above primary requirements, the mobile application should be able to:

* Enable on-field inspectors to collect and submit information to the <PROJECT> analytics platform as soon as possible
* Enable <PROJECT> to leverage the new analytics platform and access useful insights on its operations and optimize its planning



*Figure 1: Key business goals*

## Design goals

The application should:

1. Be compatible with multiple mobile platforms and devices
   1. Work on iOS, Windows Mobile and Android platforms
   2. Work on different form factors like phones and tablets
2. Support offline capabilities in mobile devices
   1. Mobile application should work even if there is no internet connection
   2. Auto-sync data captured to server once online
3. Comply with security standards
   1. Follow <CLIENT> <PROJECT> standard security guidelines of Application Development
   2. Data encryption while storing data on mobile device
   3. Mobile application should access server-side services over secured VPN
4. Be responsive
   1. Provide a user-friendly experience and handle requests with impressive UI performance
   2. Mobile application should be responsive even if there is no internet connection

# Non-functional requirements

1. **Reliability**: System will be highly reliable and less prone to failures. Mobile application should be able to work even if there is no network connectivity.
2. **Security**: System will comply <CLIENT> standard security guidelines. Also, the mobile application should allow operations through a secured VPN connection.
3. **Usability**: System will be easy to use and operate. User interface should be responsive to various form factors as well as orientation changes
4. **Portability**: Mobile application will be easily deployable on different platforms (iOS, Windows, Android).
5. **Performance**: Mobile solution must be accessible to all the inspectors and UI and solution performance should be as per the performance standards of <CLIENT>.

# Solution overview

## Key design considerations

*Figure 2: Key design considerations*

### Platform independence

Solution design will support platform agnostic deployment i.e.

* Same mobile application will be deployable on different devices with different operating systems.

### Reliability

System will be highly reliable and will be able to work in following cases of network partition-

* Mobile device is not able to communicate with <PROJECT> Backend Server because <PROJECT> Backend Server is not available.

In this case, the mobile application will work in isolation and will be able to upload the changes once the system is available.

### Security

Key design considerations from security perspective are-

* Storage Security
  + Data classification
    - Data entities will be classified based on standards and compliance requirements. Appropriate security approach will be defined and implemented based on classification. E.g.
      * Some data might be allowed to store in non-encrypted manner
      * Some data might only be allowed to store in encrypted manner
      * Some data might not be allowed to store at all
  + Data at rest
    - Any data (including files) will be stored in encrypted manner on mobile devices.
  + Data in transit
    - Any data transmitted to and from mobile application and <PROJECT> Backend Server will be done on private encrypted channel
* Application security
  + All <PROJECT> Backend APIs will be accessible to authenticated devices/users via <CLIENT>.ID authentication service

## Solution Concept

Below diagram depicts the concept of this solution:

*Figure 3: Proposed solution concept*

Mobile application will communicate with the <PROJECT> Backend Server via web services over private channel created through SSL over VPN capabilities. VPN will only allow access to <PROJECT> Backend Server APIs and <CLIENT> authentication service (<CLIENT>.ID) from the mobile application.

Solution will be composed of following key components-

* **Mobile Application**: This component represents the application deployed on various mobile devices. It will integrate with <PROJECT> Backend Server using web API.
* **Secure VPN**: This component will enable extension of <CLIENT> <PROJECT> private network using VPN over SSL connectivity with mobile devices

## Mobile Application

### Logical architecture

This section briefly explains the idea of envisioned system based on requirements and design considerations identified in previous sections.

*Figure4: Logical Architecture-Mobile POS Client Application*

The architecture of the mobile application is based on the accelerator from Nagarro. The accelerator will provide a base framework with following key highlights:

#### Mode-View-ViewModel

The internal structure of the mobile application will have components logically grouped into separate stacked layers (providing separation of concerns) that communicate with each other through defined interfaces.

The base framework will provide a solid backbone using MVVM pattern, hence the presentation layer will be composed of Views & ViewModels. The Models consist of the combination of Business and Data layer components.

Benefits:

* Isolation: Technology and functional upgrades can be isolated at specific layers.
* Testability: Different concerns in form of layers can be tested independently. The provision of ViewModel will further increases the testability of the UI layer.
* Re-usability: Lower layers having no dependencies on higher layers can be reused in other scenarios.

#### Object oriented design

The application design will be based on object-oriented principles to create self-sufficient objects, each containing the data and the behavior relevant to the object. Some of the key OOAD principles that may be considered include:

* Open-Closed Principle (OCP)
* Single Responsibility Principle (SRP)
* Interface Segregation Principle (ISP)
* Liskov Substitution Principle (LSP)
* Interface oriented design instead of inheritance oriented design
* Composition instead of inheritance

#### Cross cutting concerns

These services provide infrastructural functionalities that should be used across all modules in the application, including:

* Configuration: These services will provide support for various application level settings that may be required to be configured.
* Exception/error Handling: This is the framework for handling exceptions that occur during the application execution.
* Security: These services will provide application level security to the various functions of the application, such as data encryption and user authentication etc.
* Auditing and Logging: These services provide information logging about conditions or events, status updates, error/exception states, etc. The log store will be configurable to be simple text files or database.
* Internationalization & Localization: These services will provide the way to adapt forms in the language of user’s choice.

Based on data classification, data will either be logged, logged with masking or not logged at all.

#### Solution for key functionalities

##### Offline mode and Auto-synchronization of data

It is required that the mobile application should work even if a mobile device is not able to connect to <CLIENT> <PROJECT> Backend services. All the data which has been filled on the device will remain on the device till the time it is synced with backend.

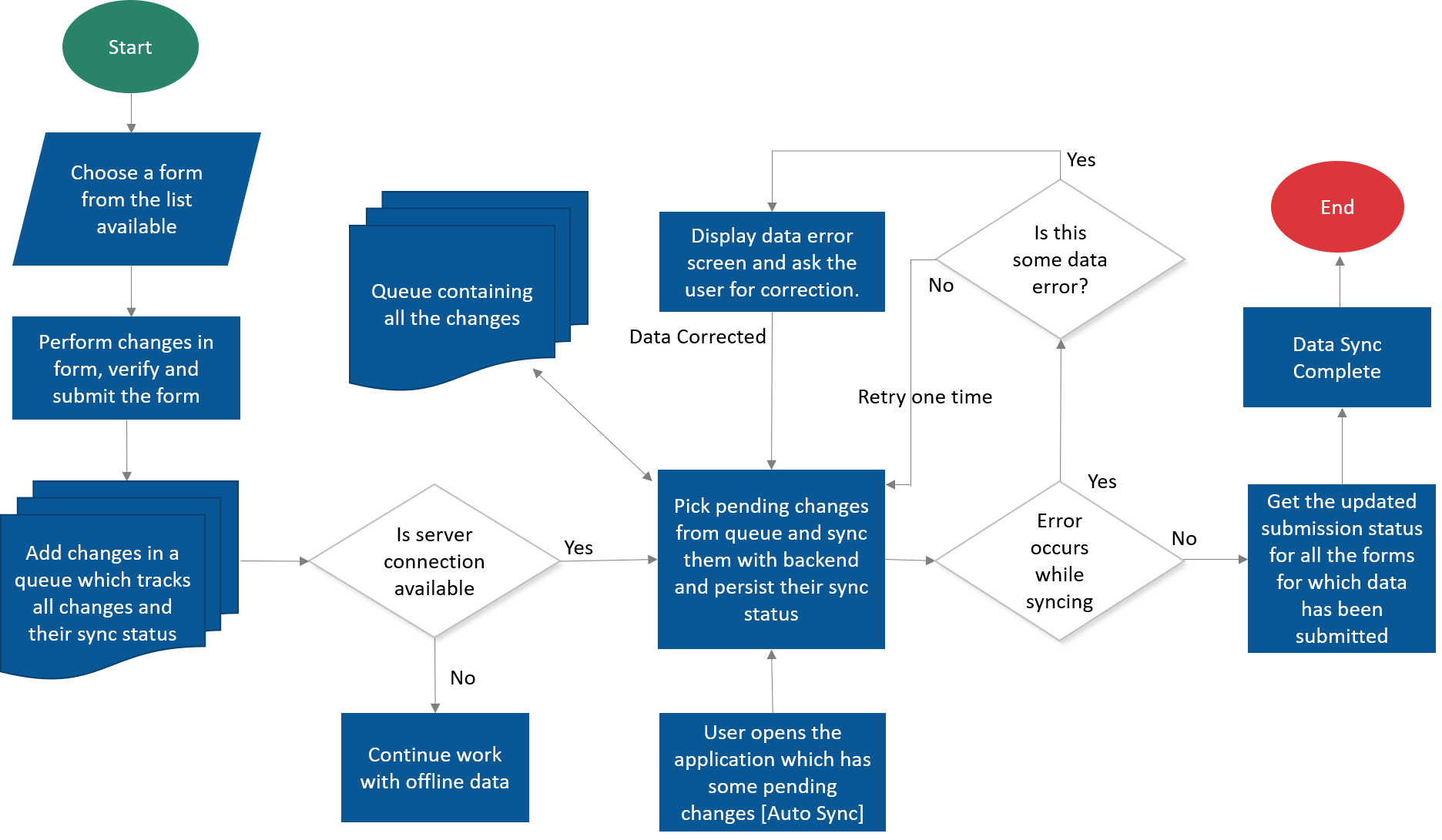
Key aspects of offline storage and synchronization are-



*Figure 5: Offline storage & Sync aspects on mobile application*

* **Persistence**: Data captured will be stored in SQLite database locally on mobile device. This database will store cases with their forms data, master data for auto-fill, and business rules. It will also keep the data synchronization status with server. The persistence strategy will have lower memory footprint.
* **Queuing**: Efficient queue management plays a key role for sync operations. A FIFO queue will be maintained in local database for changed data.
* **Conflict resolution**: Only one agent will be working on one case therefore the older information if exists for a form will be overwritten by the new information.
* **Recovery**: A strategy to seamless recover the state of data correctly during failure conditions and start it from the logical point is important. Mobile application design will take care of persisting the intermittent states of forms sections as they are filled by the agent so that no data loss occurs in case of any failure. Retry and/or circuit breaker pattern(s) will be implemented to cope up with integration issues.

Below diagram depicts an approach-



*Figure 6: Offline data sync flow*

More details about the data synchronization approach are mentioned at [this link](#_Mobile_POS_client).

#### Dashboard and Analytics Reports

The mobile application user will be able to view his dashboard and analytics reports that can be accessed by him once he logs in with his <CLIENT>.ID credentials.

For Android and iOS, the reports will be viewed by opening the report link on WebView. (Assuming Tableau). For Windows phones, the <PROJECT> Backend Server will expose APIs that will be used to create Dashboard and render reports natively. (Assuming Tableau doesn’t work on Windows phones).

If PowerBI is used, reports will be rendered using IE on Windows Phones as well.

#### Forms

The mobile application will show a list of forms that can be used on the field by an inspector. The forms will be easy to understand, fill and navigate by the inspector.

The mobile application will show 5 pre-defined forms that an inspector can choose, fill and submit.

#### Security

##### Authentication

The mobile application will integrate and communicate with <CLIENT>.ID identity service for user authentication. Key steps involved in communication with <CLIENT>.ID are:

* Step 1: Invoke the <CLIENT>.ID Authorization web service.
* Step 2: Securely store the OpenID Connect implicit flow access token.
* Step 3: After token receiving, allow users to login with supported federated identity provider.
* Step 4: Mobile app will not store <CLIENT>.ID service account details.

###### Offline authentication

To support offline authentication on mobile device, user credentials will be stored in the credential store for the respective mobile environments. In case the user is offline then the credentials which are saved in credential store will be used for authentication. The passwords will be hashed before storage. Offline authentication will work only for those users who have logged once successfully in online mode on the device.

*Figure 9: Offline authentication flow*

##### Encryption

Data encryption will be done by leveraging SQLite SQLCipher library. SQLCipher provides transparent 256 Bit AES encryption of database files which satisfies <CLIENT> <PROJECT> encryption standard of having minimum 128-bit encryption. The credentials stored for offline authentication will also be stored in encrypted mobile platform specific files like Keystore for Android, Keychain for iOS and Credentials locker for the Windows mobile platform.

#### Integrations

##### Secure VPN Connection

All communication between mobile application and the <PROJECT> Backend server will take place on VPN over SSL. In this manner, <CLIENT> <PROJECT> Backend private network will be extended to mobile application access outside <CLIENT> <PROJECT> Backend premise over the internet. It is assumed that device level VPN connectivity will be provisioned with mobile devices to connect <CLIENT> <PROJECT> premise.

# Technology

Below is the proposed technology for the solution-

## Mobile Application

|  |  |  |
| --- | --- | --- |
|  | Aspect | Recommendation |
| 1. | Programming paradigm | Object oriented |
| 2. | Architecture style | Layered architecture |
| 3. | Language | C# |
| 4. | Service communication | HTTPS based, RESTful styled |
| 5 | Data exchange format | * JSON |
| 6. | Mobile UI Development | * Xamarin.Forms * Xamarin.iOS * Xamarin.Android * Xamarin.UWP |
| 7. | Database | SQLite |
| 8. | Database encryption library | SQLCipher (256-bit Encryption) |
| 9 | ORM | SQLite.Net |

# Assumptions

Following assumptions are made to devise the proposed solution approach.

1. Data for dashboard will be fetched from a single endpoint only.
2. System will not support dynamic form creation. If an existing form needs to be updated or a new form needs to be added, the mobile application has to be updated with new forms module.
3. Implementation of only 5 data capture forms has been considered.
4. On an average, a form will have 25 different fields for data capture.
5. The data sync time & performance will depend on the amount of the data to be synced, network bandwidth and backend systems performance.
6. For tablet and iPads both landscape and portrait mode will be supported. For iPhone, android and windows smartphones only portrait mode will be supported.
7. Form data validation (if any) will take place only on the mobile application.
8. Business continuity requirements and solution for application/data and disaster recovery will be discussed during project execution.
9. Device level VPN connectivity will be provisioned with mobile devices to connect <CLIENT> <PROJECT> Backend premise.
10. Sync operation for submitting forms data will be performed when the application is in the foreground mode only. As a best practice, mobile platforms recommend not to run memory intensive tasks when in background, and the operating system can kill such application in case other apps in foreground need memory.
11. Application will only display forms preconfigured on the mobile application.
12. All documentation, exception handling, logging messages etc. in English language only.
13. VPN configuration on mobile devices and server will be done by <CLIENT> <PROJECT> team.
14. Signatures will be captured as image by mobile application. Further, application will not have any validation support for them.
15. Google Maps/Bing will be used to display routes, but no navigation will be provided within the mobile app. For navigation functionality, user will be redirected to Google Maps/Bing official mobile app for navigation from current location to target location.
16. Maps will work only in online mode and routes and locations will not be shown in offline mode.
17. If a device containing offline data is lost, there will be no way of retrieving the captured information.
18. Any changes required in the existing systems will be undertaken by the <CLIENT> <PROJECT> Backend team.
19. All mobile application testing will be done on a minimum of 2 Mbps 3G connection.

# Scope considerations

Following items are kept in consideration for scoping perspective.

## In-scope

Following are in scope items-

1. Design and development of the <PROJECT> Now mobile app using Xamarin, supporting iOS, Android and Windows platform for both phones and tablets.
2. Authentication using <CLIENT>.ID.
3. Support of iOS v10.0+, Android v5.0+ and Windows 10 mobile/tablets
4. Communication of the mobile app with <CLIENT> <PROJECT> Backend server through SSL over VPN.
5. Testing includes manual functional testing, system integration testing, automated regression testing, performance testing, security testing, multi-device testing, and usability testing.
6. Testing of mobile apps on 3 devices for each platform (iOS, Android and Windows 10), specified by <PROJECT>.
7. Crash analytics, beta testing and usage tracking using “HockeyApp” platform.

## Out scope

Following are out of scope items:

1. Dynamic forms creation which can be updated from the server.
2. Key generation, storage and rotation solution for data encryption.
3. Accessing <CLIENT> <PROJECT> Backend premise services over application level VPN connectivity offered by Secure VPN.
4. Email & push notifications.
5. Language translation.
6. Data level authorization.
7. Anything not mentioned in-scope.