Technical Specifications

1. Overview

The Anti-Corruption Commission (ACC) wish to develop a mobile application which would ease the performance of its duties. The full system to be delivered must consist of the following components:

1.1. A thin-client mobile application (or "the mobile app"),
1.2. A secure RESTful server which interacts with the thin client (or "the server").
1.3. A secure RESTful API provided to the mobile app by the ACC (or "the API"). The developers will need to use the API on their server.

2. Technologies Required

The following technologies must be used in the system's implementation. Other libraries and technologies may be used on top of those prescribed in this section.

2.1. The **mobile app** for the mobile application **must** use a cross-platform mobile development framework such as Xamarin, Ionic, Flutter, or React Native.
2.2. The **server** must be built with either .NET (C#), Express (NodeJS), Laravel (PHP), CodeIgniter (PHP), OctoberCMS (PHP), or Echo (GoLang).
2.3. PostgreSQL must be used for data storage
2.4. Open-source libraries for authentication and authorization, e.g. PassportJS, Authboss, etc.
2.5. The server must be hosted on an international cloud platform such as DigitalOcean or Azure; acquiring and configuring a server will be handled by the developers after conferring with the ACC.

3. Non-Functional Specifications

3.1. The UI must closely resemble the images provided in this document.
3.2. The UX must be intuitive and helpful.
3.3. The mobile application must be secure.
3.4. The mobile app must use the device's battery efficiently.
3.5. The system must ensure that the user's private information is cryptographically protected.
3.6. The user interface must be in Dhivehi and must display Thaana with a formal typeface (i.e. a typeface considered suitable for official government documents).
3.7. Offline performance of the mobile app must be possible; each action made by the user on the mobile app while offline must be synced with the server once access to the Internet is restored.
3.8. Users will be divided into three main roles: Superadmin, Admin, and Public. Role-based user authentication and authorization should be handled by the thin mobile app.
3.9. All calls to any APIs exposed by the ACC will be protected by a security token which will be included in the mobile app.
3.10.   Extensive documentation is mandatory; it is expected of the developer that methods and functions are sufficiently commented, and that documentation regarding the behaviour of the entire system and its components is provided.
3.11.   The production process must undergo extensive unit-testin

4. User Roles and Privileges

4.1. The Superadmin will be able to create and delete and ban Admin and Public users, and will have all the privileges of Admin and Public users.

4.2. Public users can be either authenticated or anonymous (unauthenticated).

5. Functional Requirements

5.1. The mobile app should be able to securely register, log-in, and log-out Public users.

5.2. The mobile app must display a main navigation menu as shown in Fig.1. to all Public users (authenticated and anonymous) on loading the app. A **loading-page** is optional. The navigation menu must implement the following behaviour:

5.2.1. Fig. 3 will be reached by clicking "ޮޮޮޮޮޮޮޮ"

5.2.2. Fig. 7 will be reached by clicking "ޮޮޮޮޮޮ ޮޮޮޮޮ"

5.2.3. Fig. 8 will be reached by clicking "ޮޮޮޮ"

5.2.4. Fig. 9 will be reached by swiping left

5.2.5. Fig. 10 will be reached by clicking "ޮޮޮޮޮޮ ޮޮޮ"

5.2.6. Fig. 14 will be reached by clicking "ޮޮޮޮޮޮ"

5.2.7. Fig. 13 will be reached by clicking "ޮޮޮޮޮޮޮ"

5.3. *Admin Panel*

5.3.1. **The server** must provide a simple graphical user interface to the Admin and Superadmin users. All operations pertaining to the Admin users shall take place through this graphical user interface. A Bootstrap-based framework like AdminLTE may be used.

5.4. *Access*

5.4.1. **Registration**

5.4.1.1. **The mobile app:** An unauthenticated Public user is able to register themselves on the system (see Fig. 2.)

5.4.1.2. Registration and authentication of users should be handled on the server and must not be delegated to an external platform. The following data are required:

5.4.1.2.1. The user's full name in Thaana and Latin,

5.4.1.2.2. The user's National Identity Card number,

5.4.1.2.3. User's email address,

5.4.1.2.4. User's phone number,

5.4.1.2.5. (Optional) a profile picture. In the absence of a profile picture, a default placeholder profile picture must be used.

5.4.1.3. **The server:** The server creates a new User and is viewable through the Admin Panel by the Superadmin and the Admin Panel. The User's name, address, contact information, and scan of NID card will be viewable. **Registration** can only be completed when either a Superadmin or an Admin manually authorizes it through the **server.** The user will receive a notification of this final authorization.

5.5. Login

5.5.1. **The mobile app:** The mobile app an unauthenticated user is able to log in to the system (see Fig. 2.) A reasonable password reset module should be implemented.

5.5.2. **The server:** The server handles authorization and authentication.

5.6. *User Navigation Menu*

5.6.1. **The mobile app** will provide a main menu for authenticated Public users as shown in Fig. 4. Each of the links provided in Fig. 4 will lead to the relevant interface.

5.6.2. **Fig. 5** *will be reached by clicking* "ﺩﻬ‍ﺳﻮ ﻫﺮﺭﻮﻮﻥ"

5.6.3. **Fig. 6** *will be reached by clicking* "ﺩﻬﺳﻮﺭﺉ ﺳﻴﻴﻘﺲ ﻗﻮﻮﻥ"

5.6.4. **Fig. 11** *will be reached by clicking* "ﺍﻮﻣﺮﺳ‍ﻭﻗﺴﺮﻗﺮ ﺍﻮﺩﻮﻥ"

5.6.5. **Fig. 12** *will be reached by clicking* "ﻛﺮﺩﺉ ﺑﺮﻗﺉ"

5.7. *Investigation Module*

5.7.1. Submitting an Issue

5.7.1.1. **The mobile app:** An authenticated Public user should be able to fill out a form as per Fig. 5 which will be received by the **server.** The user will be able to upload **images, videos, audio recordings,** and **PDFs.**

5.7.1.2. **The server:** The server will receive all submitted information about the case. The case will be given a **tracking ID**.

5.7.2. Viewing a Submitted Issue

5.7.2.1. **The mobile app:** An authenticated Public user will be able to view all the issues that they had previously submitted in one interface as per Fig. 6.

5.7.2.2. **The server:** Admin users will be able to view all issues submitted through the mobile app. Admin users will be able to view the **tracking ID** of the case.

5.7.3. Check Status

5.7.3.1. **The mobile app:** An authenticated Public user will be able to check the status of a case by submitting the case number.

5.7.3.2. **The API:** The API will receive a string containing the case number, the user's name, and phone number and will provide an update of the case's latest status.

5.8. Whistle-blower Module

5.8.1. Whistleblowing:

5.8.1.1. **The mobile app:** The mobile app should contain a **secure** interface which will allow an unauthenticated (anonymous) Public user to **anonymously** upload **images, videos, audio recordings,** and **PDFs.** As per Fig. 7, the user will be required to write out a summary of the issue.

5.8.1.2. **The server:** The server will receive this submission and send a status message to the client mobile app.

5.8.2.View Whistleblowing:

5.8.2.1.   **The server:** The Admin Panel will contain a clickable list of all the issues submitted to the ACC through whistleblowing. On clicking each item, the Admin user will be able to access the submitted information.

5.9. *Content*

5.9.1.   Create, Read, Update, and Delete Content Items

5.9.1.1.   **The server:** The Admin user will be able to perform CRUD operations on Content-Items through the Admin Panel. Content-Items here refers to all written text which is contained in posts and static pages (e.g. news-items, announcements, etc.). Admin users will be able to handle **text,** an associated **cover image,** and also **PDF documents** to any given **Content Item.** All Content Items will be viewable to the Public user.

5.9.2.   Read and download content

5.9.2.1.   The mobile app: The Public user will be able to read written text and download **PDF** documents that have already been uploaded to the **server** by the Admin. Breaking News items or Important Announcements will be sent to the user by a push notification. See Figs 9 and 8.

5.9.3.   Important and High-Priority Content-Items

5.9.3.1.   The mobile app: The *latest and highest-priority* Content Items will be displayed at the top of the **Interface** shown in Fig. 9.

5.10.   *Request an Appointment*

5.10.1. Request a Member Appointment

5.10.1.1.   The mobile app: An authenticated Public user will be able to **request** an appointment with a Member of the ACC through the app.

5.10.1.2.   The API: Once a submission is made, a PUSH request will be sent to the API exposed by the ACC. On successful processing, the server will respond with a positive status message.

5.10.2. Check Status of Member Appointment

5.10.2.1.   The mobile app: An authenticated Public user will be able to **view** the status of their requested appointment with a Member.

5.10.2.2.   The API: On view, a GET request is sent to the API exposed by the ACC which will return a JSON string indicating the status of the appointment request.

5.11.   *Criminal Record*

5.11.1. Request Criminal Record

5.11.1.1.   The mobile app: An authenticated Public user will be able to **view** their criminal record, if any, on an interface resembling Fig. 12.

5.11.1.2.   The API: Once a submission is made, a PUSH request will be sent to the API exposed by the ACC. On successful processing, the server will respond with a positive status message.

5.12.   *Comments*

5.12.1. Send a Comment

5.12.1.1.   The mobile app: A Public user (authenticated or anonymous) can send a comment through a simple form (see Fig. 13).

5.12.1.2.   The server: The text of the Public user's submission will automatically be sent by email to info@acc.gov.mv.

5.13.   *Analytics (OPTIONAL)*

5.13.1. Collect data

5.13.1.1. The server: The server will collect information about users on an hourly basis. Information would include the Public user's latitude and longitude, and a Boolean value indicating if the user has accessed the app within the hour or not.

5.13.2. Data Analytics

5.13.2.1. The server: Allow the Admin to export all the data to a CSV file. Also, provide simple analytics across all users indicating the average locations of the most frequent users.

5.13.3. Data Visualization

5.13.3.1. The server: The Admin user must be able to view the data and performance of the mobile app through the Admin panel.

6. Production

The development team will assist the ACC in the deployment of the server and the mobile app.

The development team must also share the source code through a private Git repository with the ACC.

7. Deployment

The ACC will **own** the source code for both the server and the mobile app, and must therefore have full access to and all privileges over the source code that has been deployed into production.

The developers **must** provide the ACC with assistance with deployment, debugging, and other related purposes for a maximum period of 6 months after delivery.

8. Technical Evaluation

Successful candidates will be evaluated on the condition that their proposals pass our technical evaluation stage. Candidates who submit proposals of inferior quality which do not satisfy our requirements will be disqualified from the bidding process.

The proposals will be evaluated based on criteria described in Section 9.

The successful candidate must be able to fulfil more than 50% of the total number of points listed in Section 9.

9. Technical Evaluation Criteria

There are 5 **optional** points in addition to a total of **17** points. Therefore, a successful candidate must score **9** points or higher in this Section in order to proceed in the bidding process.

Note: *Mandatory* points are written in the form (1 pt) and *optional* points are written in the form (+1 pt), and an ***obligatory penalty*** is written in the form (-1 pt).

9.1 The proposal must be written in clear English, free of major grammatical errors (1 pt.)

9.2 The proposal document must contain **all** of the following sections: (1 pt.)

　9.2.1 Section with a table of contents (1 pt)

　9.2.2 Section with **overview/summary** of the full system architecture (1 pt)

　9.2.3 Section with **overview/summary** of the data schemas to be used (1 pt)

　9.2.4 Section with **detailed** descriptions of the modules (1 pt)

9.2.4.1 All of the modules to be written to achieve Sections 9.2.2 – 9.2.5 must be described specifying their names, types (i.e. whether they are a **function, class, interface,** etc.), their parameters, and a short description of their purposes/functions. (1 pt)

9.2.4.2 A set of **Sequence Diagrams** are present which illustrate explanations of the more complicated behaviour implemented. The choice of which behaviours to highlight will be left to the proposal's author.(+1 pt)

9.2.4.3 The relationships between the **modules** must be labelled appropriately. (1 pt)

　9.2.5 Section with a **detailed** description of the database schema. (1 pt)

9.2.5.1 **Optionally,** use entity abstraction for the data schema. (+1 pt)

9.2.5.2 Section should include an Entity-Relationship Diagram (1 pt)

9.2.5.2.1 The diagram is annotated or is explained thoroughly. (1 pt)

　9.2.6 Section detailing miscellaneous other technical choices must be present (1 pt)

　9.2.7 Price quotation section is present. (1 pt)

9.3 The price quotation must be broken-down into subcomponents. These subcomponents **must** include the mandatory (but not necessarily the optional) subcomponents in the following list: (1 pt)

　9.3.1 Total GST (if applicable) (+1 pt)

　9.3.2 Total cost of labour (1 pt)

　9.3.3 Server hiring, set-up, and configuration costs (1 pt)

　9.3.4 Licensing fees (if any) (+1pt)

　9.3.5 Other costs (if any) (+1pt)

9.4 A price quotation which reflects a **major deviation** from general market rates will incur a penalty. **(-5pt)**

9.5 A schedule should be provided as a Gantt chart, ranging from the beginning of development to the completion of Beta-testing. (1 pt)
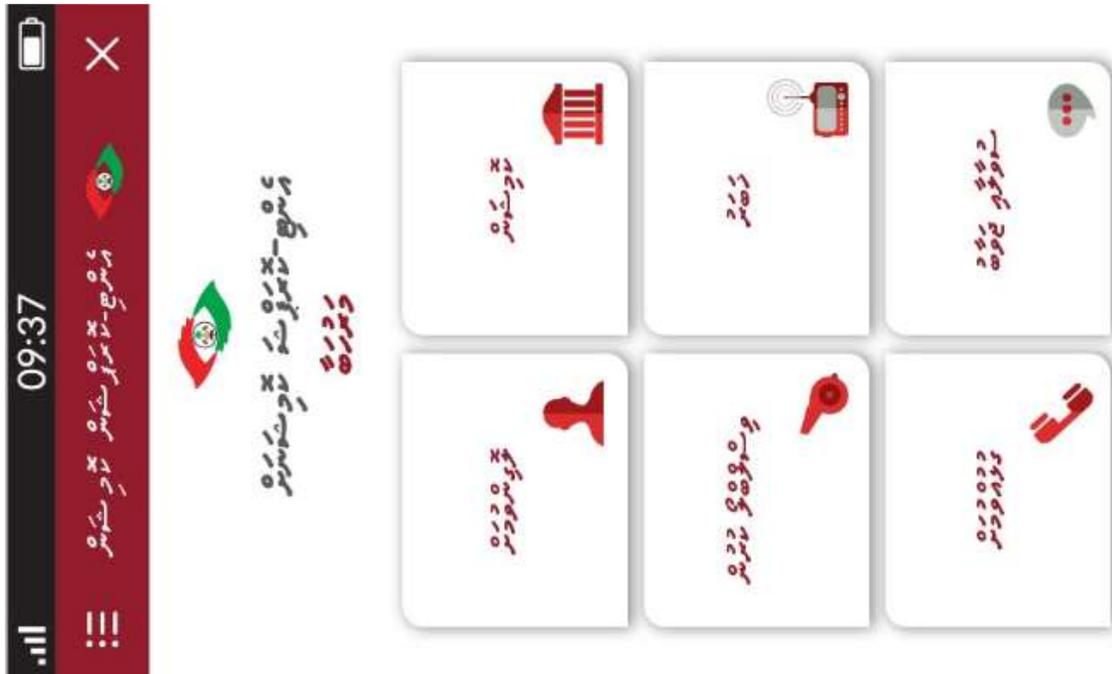

Fig 1 Navigation Menu (Welcome) [Image rotated to conserve space in document]
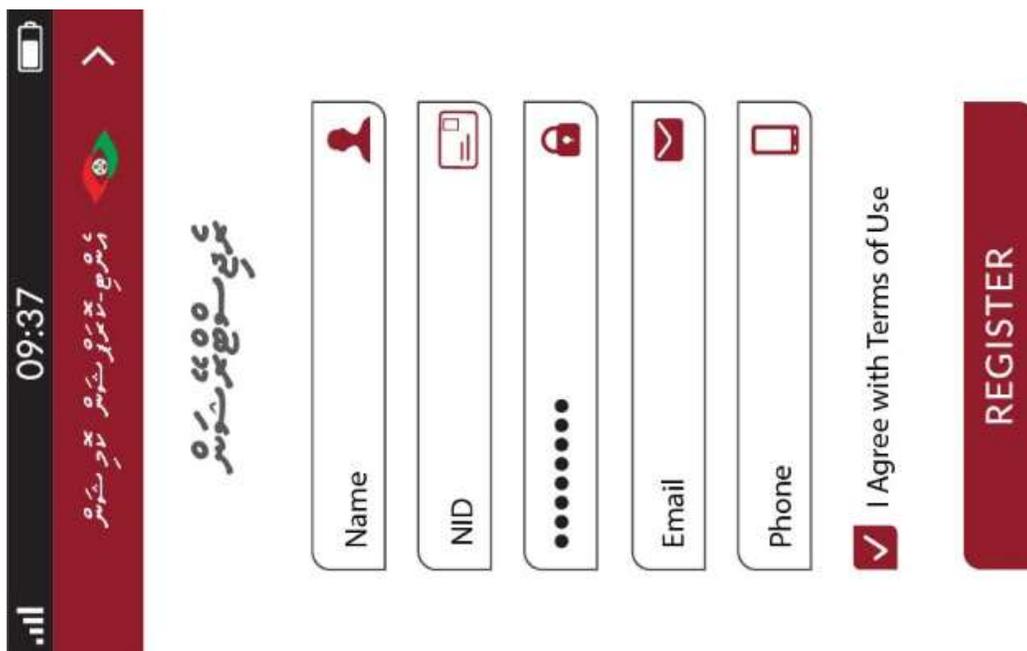

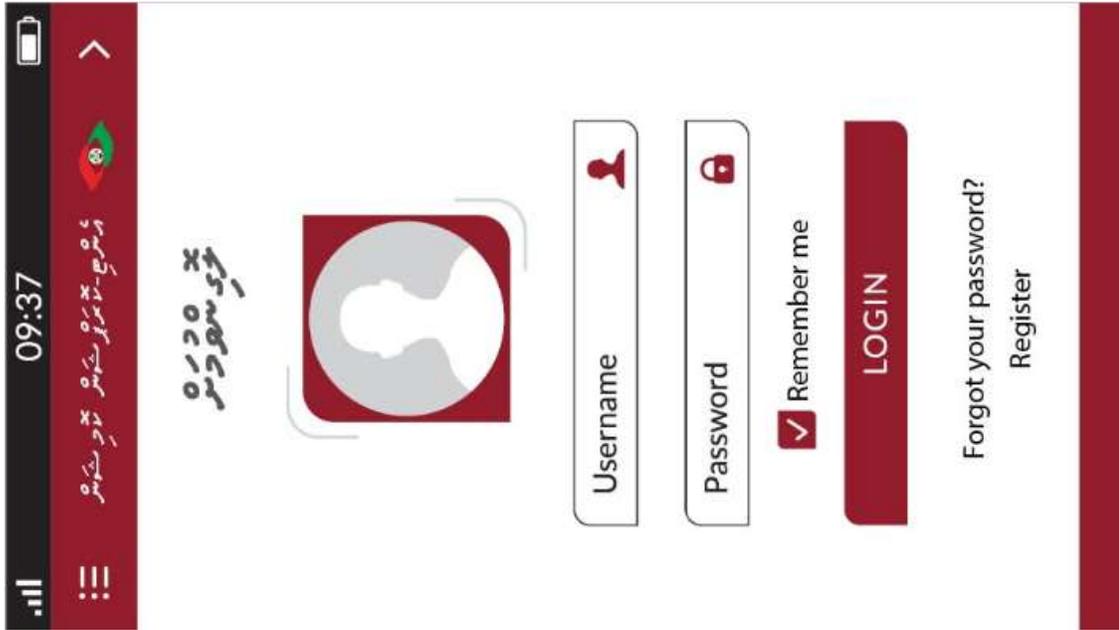*Fig 2 Registration Interface [Image rotated to conserve space in document]*

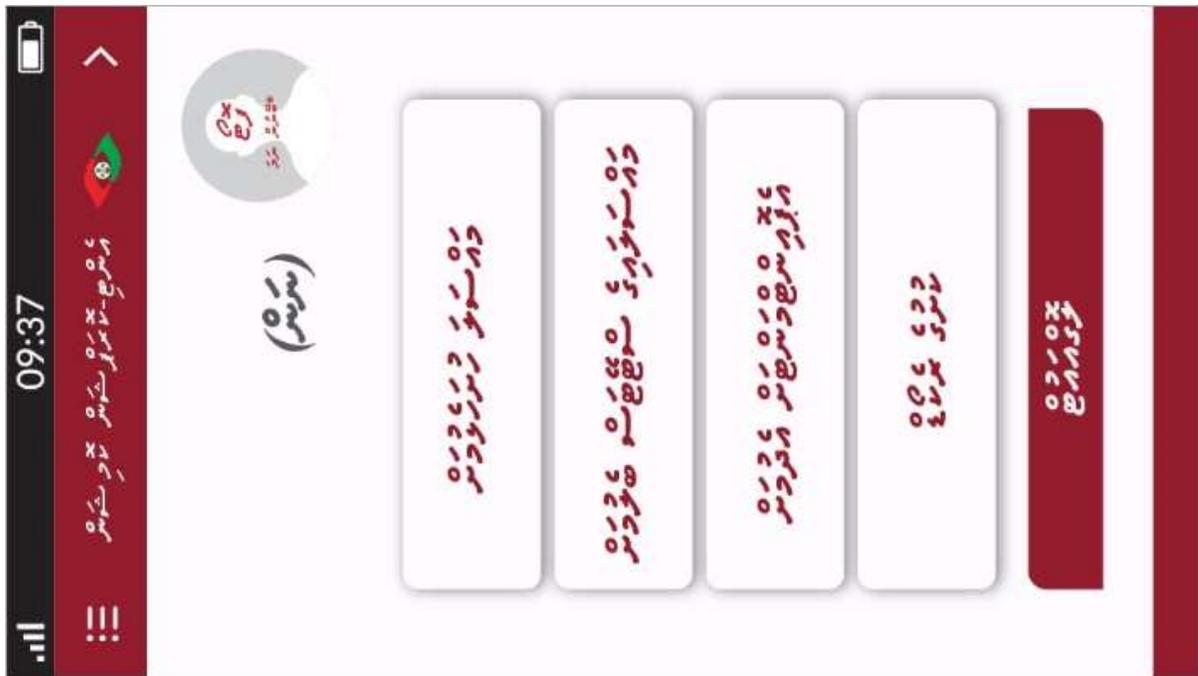*Fig 3 User Login Interface[Image rotated to conserve space in document]*

Fig 4User Menu Interface [Image rotated to conserve space in document]

123-A/IUL/2019/40


Fig 5 Issue Submission Interface [Image rotated to conserve space in document]

Fig 6 Issue Status Check [Image rotated to conserve space in document]

123-A/IUL/2019/40

Fig 7Whistleblower Interface [Image rotated to conserve space in document]

Figure 8 News Scroll Interface [Image rotated to conserve space in document]

Figure 9Latest Updates Interface (Welcome) [Image rotated to conserve space in document]
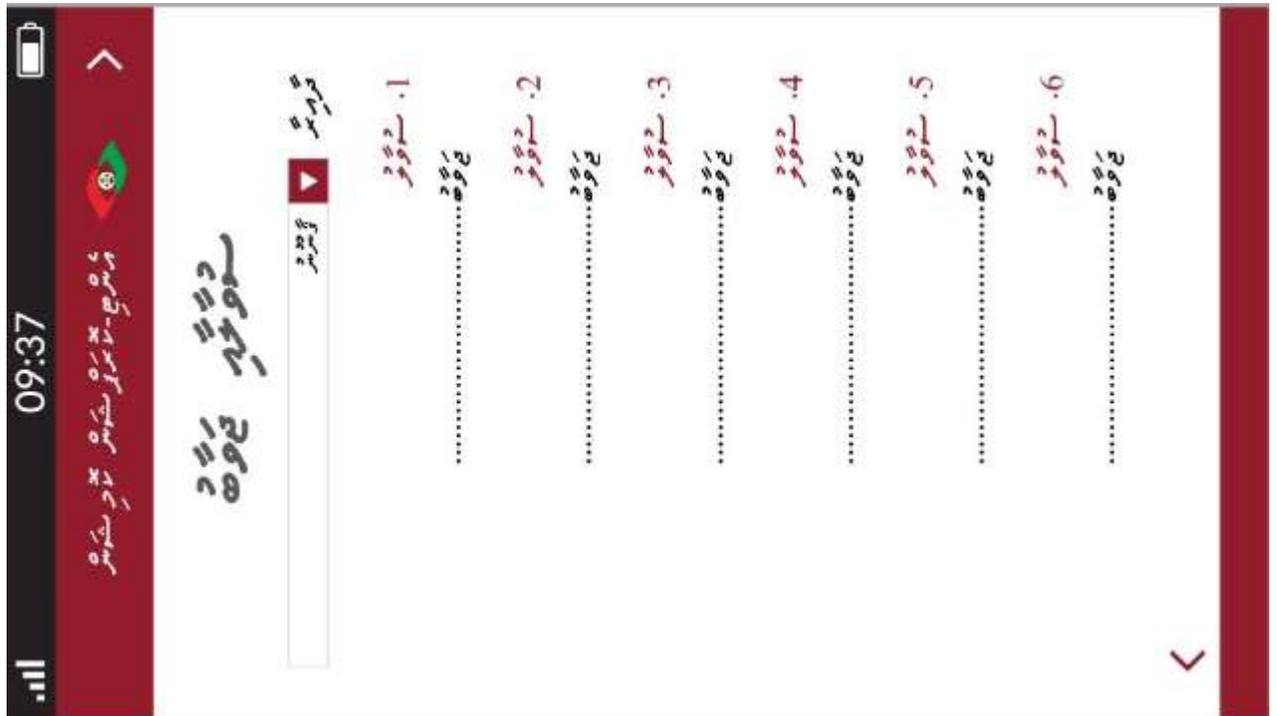
Figure 10 FAQ Interface [Image rotated to conserve space in document]

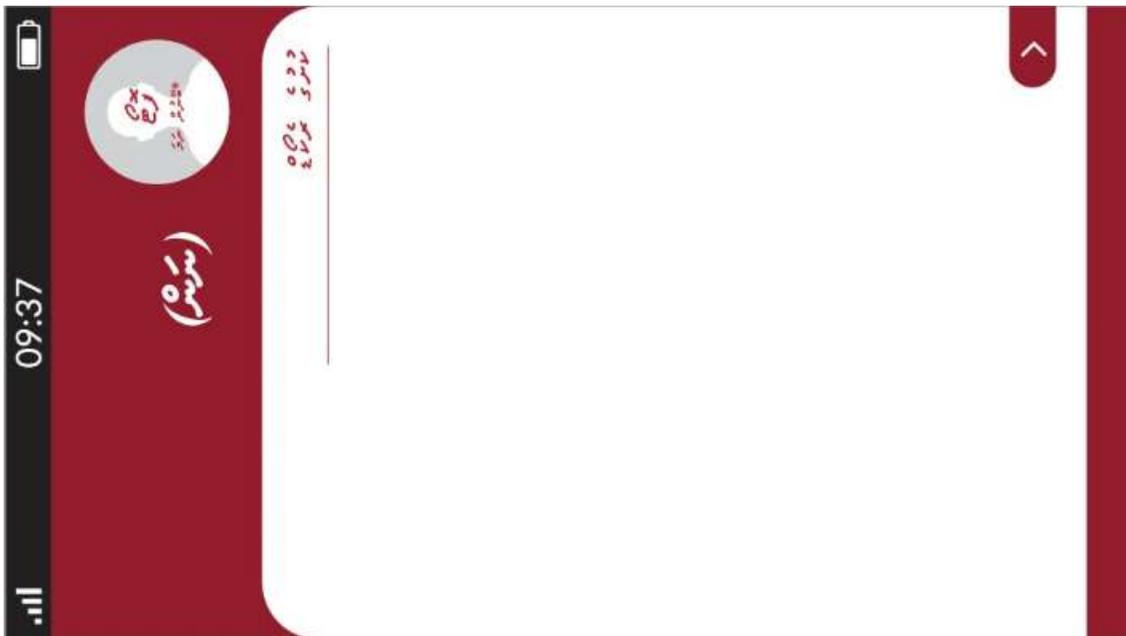*Figure 11Appointment Request Interface [image rotated to conserve space in document]*

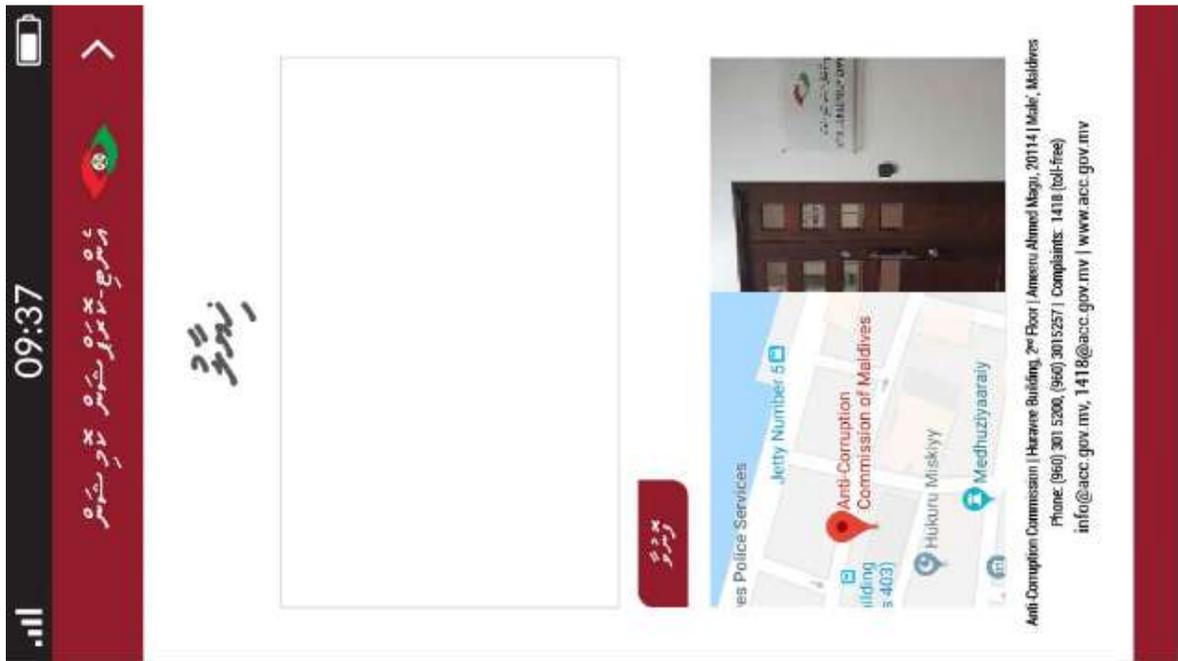*Figure 12Criminal Record Interface [Image rotated to conserve space in document]*

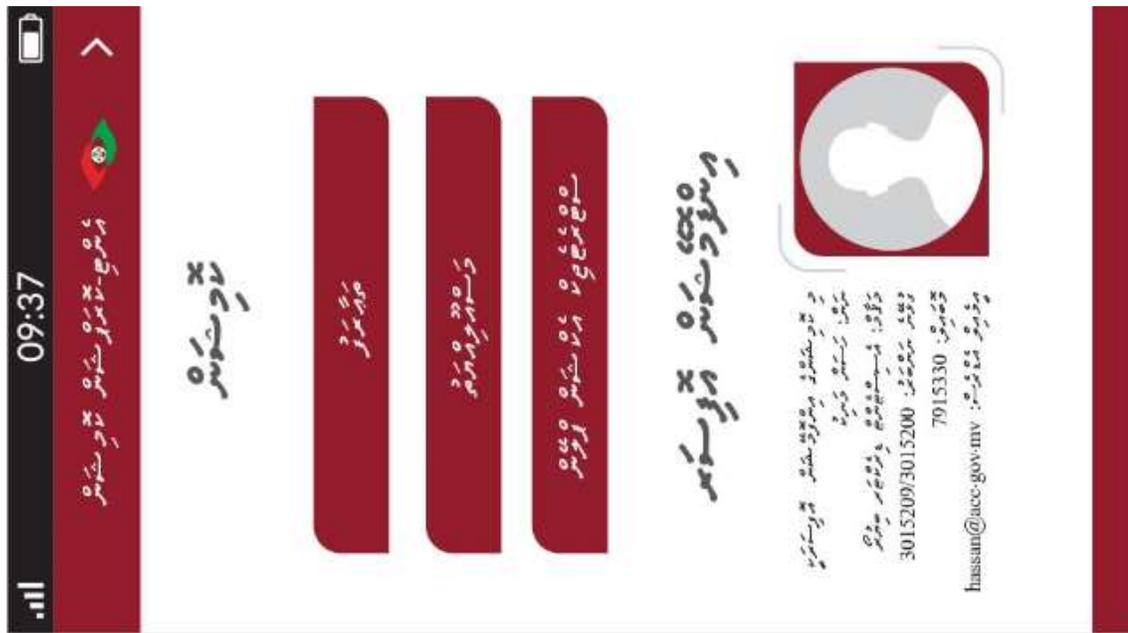*Figure 13Send Comment Interface [Image rotated to conserve space in document]*

Figure 14 Commission Information View [Image rotated to conserve space in document]