HRDK x Samsung Skills Invitational Friendship Challenge 2024

Mobile Applications Development (Skill 08)

Module A Functionality

Contents

Introduction

Due to the large number of competitors in the WorldSkills Lyon 24S, participants will be distributed across different hotels in addition to the venue. As a competitor in mobile applications development, you have been invited to create a facility inspection app for the competition staff. This app will help patrol personnel verify and report the conditions of various venues to ensure the smooth conduct of the competition. This document will describe the relevant details.

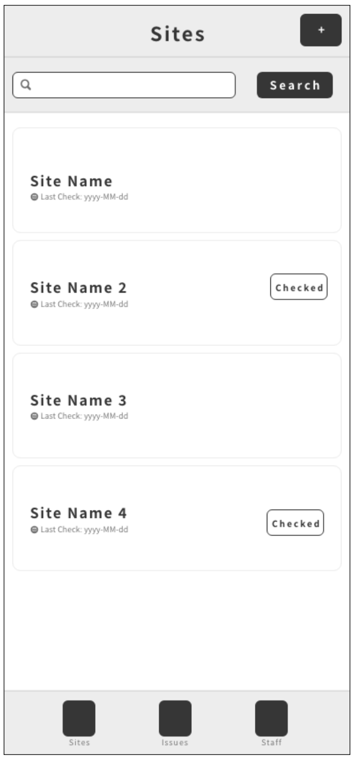
The designer has provided you with the wireframes. As a mobile application developer, your task is to develop the corresponding application based on the wireframe and the given functional requirements.

Description of project and tasks

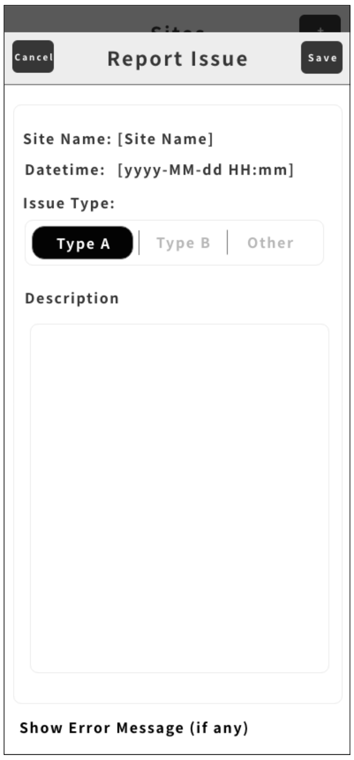
Module	Name	Competition Time	Marking Devices
A	Functionality (phone)	2.5h	Emulator： iOS: iPhone 8 Plus (iOS: 16.4) / iPhone 14 Plus (iOS: 16.4) Android: Pixel 2 (API level: 31)

Below is an overview of some wireframes and functionalities. Please refer to the files/ folder for detailed information.

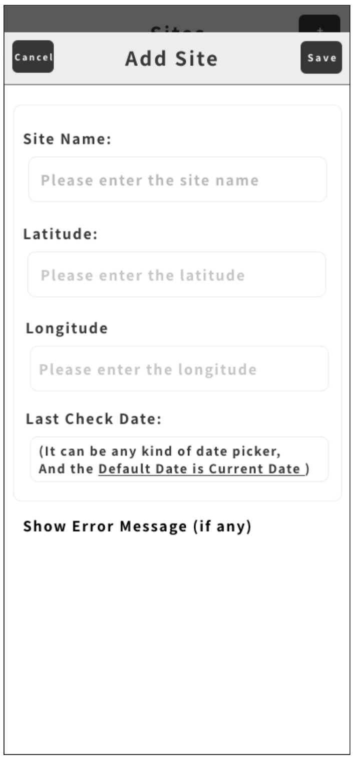
Application wireframes



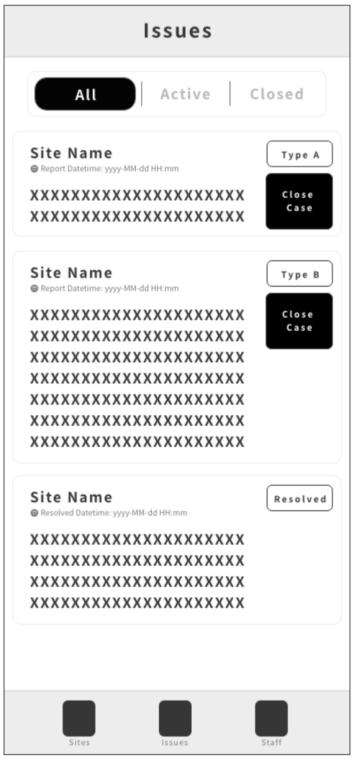
Inspection Site Listing Page



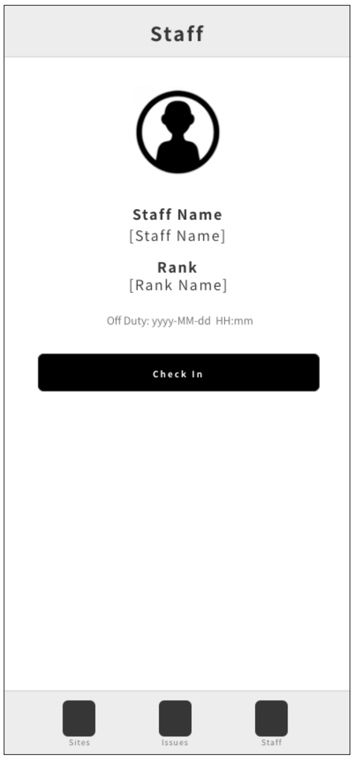
Report Issue Page (Popup)


Cancel
Add Site
Save
Site Name:
Please enter the site name
Latitude:
Please enter the latitude
Longitude
Please enter the longitude
Last Check Date:
(It can be any kind of date picker, And the Default Date is Current Date)
Show Error Message (if any)

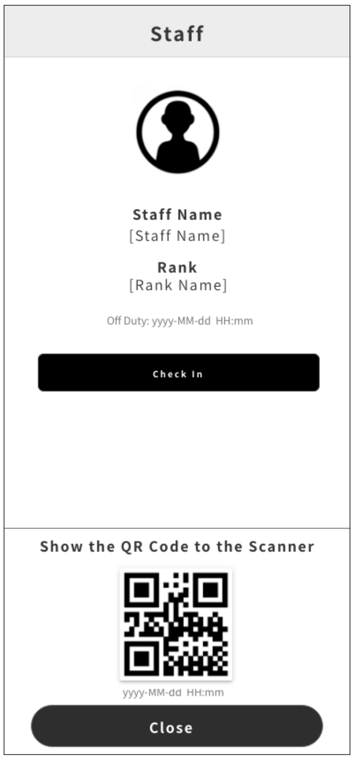
Add Site Page (Popup)



Issue Listing Page



Staff Profile Page

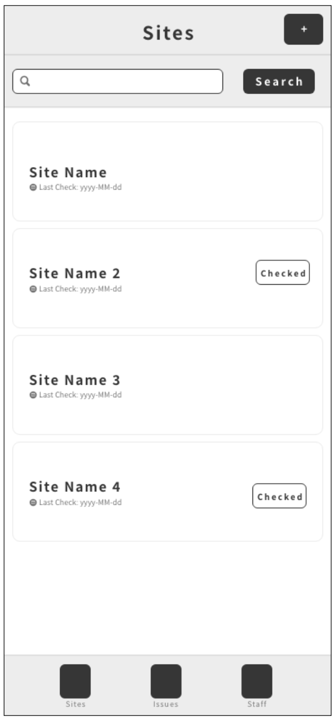


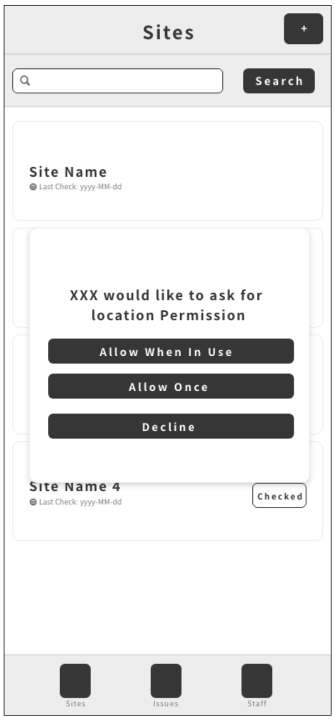
QR Code Page for Checkin / Checkout

General Demands

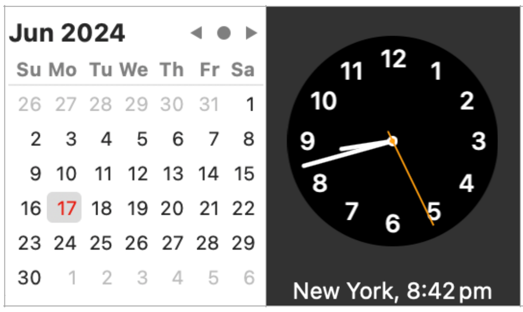
[6.1]

1. Finish the function of this application according to the demands.
2. All text with quotation marks must be the same as the *elements include*. It is acceptable have variation in the use of capital / small letter, with or without colon. (E.g.: **"My Name:"**, it can be written as "*My name*", "*my name*", *or "my Name:* ")
3. The package name / bundle identifier is expected to contain "**xx**.com.samsungskills", **xx** means your country code in small letter. *(e.g., China: cn, Japan: jp)*
4. The application should show the application icon (media-files/appIcon.png) on the home screen.[0.1]
5. All of the pages should contain a navigation bar at the bottom. (Except popup screen). [0.3]
   (a) Button navigation bar should include:
      1) "Sites" button, click to navigation **Site Listing**
      2) "Issues" button, click to navigate **Issue Listing**
      3) "Staff" button, click to navigate **Staff Page**
6. All navigation bar items should come with an icon. [0.3]
7. The first page (for a fresh start) must be **Site Listing** Page. [0.1]
8. The app should prompt a system dialog for requesting permission for retrieving user location at the first launch. **You can assume that the user will accept the request.** [3.0]
9. All data should be stored locally and could be displayed properly after restarting the app.
   a. site listing [0.8]
   b. issue listing [0.8]
   c. the check-in, check-out status of the staff [0.4]
10. Submit your work to the given git repositories. [0.3]





<u>Datetime format explains:</u>



A 24-hour date-time [yyyy-MM-dd HH:mm] format is used. The desired output of the date-time on the left would be:

[yyyy-MM-dd HH:mm]
2024-06-17 20:42

[yyyy-MM-dd]
2024-06-17

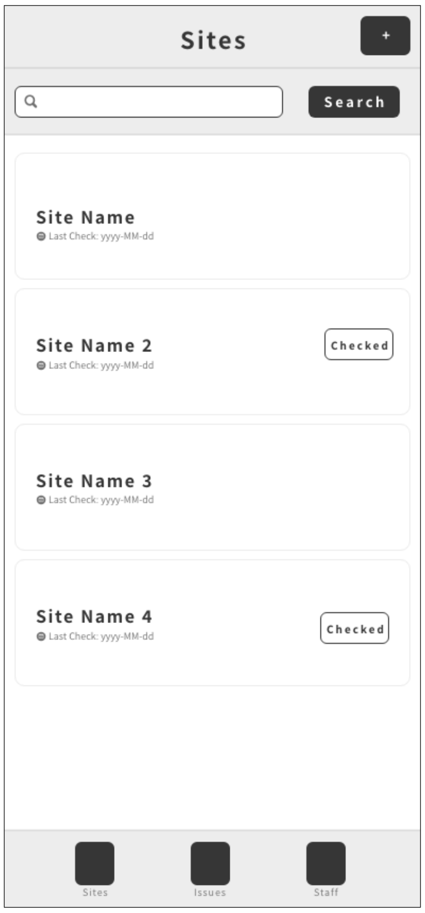
[HH:mm]
20:42

Pages Demands

1. Inspection Site Listing Page

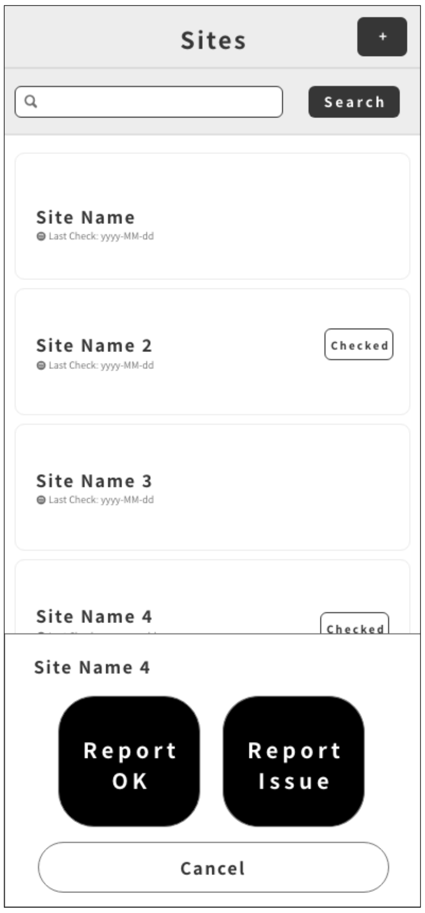
1.1 Elements include: [1.6]

1. Navigation Bar [0.1]
   1.1 **"Report Issue"** title [0.1]
   1.2 **"+"** button [0.1]
2. SearchSe Bar [0.1]
   2.1 Text input for Search Text [0.1]
   2.2 **"Search"** button [0.1]
3. Site List [0.1]
   3.1 Site cell [0.1]
   3.2 Site name label [0.1]
   3.3 Last check datetime label [0.1]
   3.4 **"Checked"** tag [0.1]
4. Site Popup [0.1]
   4.1 Site name label [0.1]
   4.2 "**Report OK**" button [0.1]
   4.3 **"Report Issue"** button [0.1]
   4.4 "**Cancel**" button [0.1]



1.2 Functional requirements: [8.7]

1. Site Listing:
   a. The list shows the preloaded data from *data/sites.json* at first launch. [2.0]
   b. Sort the list in ascending alphabetical order. [0.5]
   c. The list cell shows the **"Checked"** tag when the current datetime is less than 24 hours from the last check date. [1.0]
2. Filter the site list by site name after inputting the query string and clicking the **"Search"** button. [1.0]
3. Inspection pop-up dialogue:
   a. When the user's location is **about 100 meters** of any site, and the **site is not checked**, it will pop up the report dialogue as shown. [3.0]
   b. When the user **clicks on any site** in the site list, and the **site is not checked**, it will pop up the report dialogue as shown. [0.1]
   c. The dialogue shows the name of the selected site. [0.2]
   d. When the user clicks **"Report OK"**, the dialog will dismiss, the site's last check date will be updated to the current datetime, and the list will reflect the update. [0.2]
   e. When the user clicks **"Report Issue"**, the dialog will dismiss and pop up the "Report Issue Page" [0.1]
4. If the user clicks **"+"**, the app will pop up the **"Add Site Page"**. [0.1]
5. Design the flow and implement the deletion of a site. [0.5]

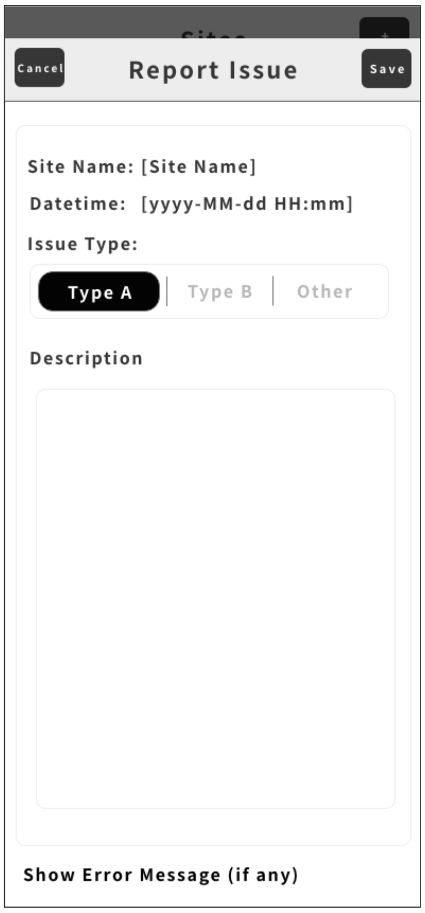


```
Test Coordinates for the GPS features:
Kopster: 45.767413686813704, 4.983290569995726
Hotel Lyon Ouest: 45.788487869947296, 4.814955535547079
Mercure Lyon Centre Lumiere: 45.74705204652239, 4.867681509372805
Appart City Lyon Part Dieu Garibaldi: 45.753766854502295,
4.854816999007168
```

2. Report Issue Page

2.1 Elements include: [1.0]

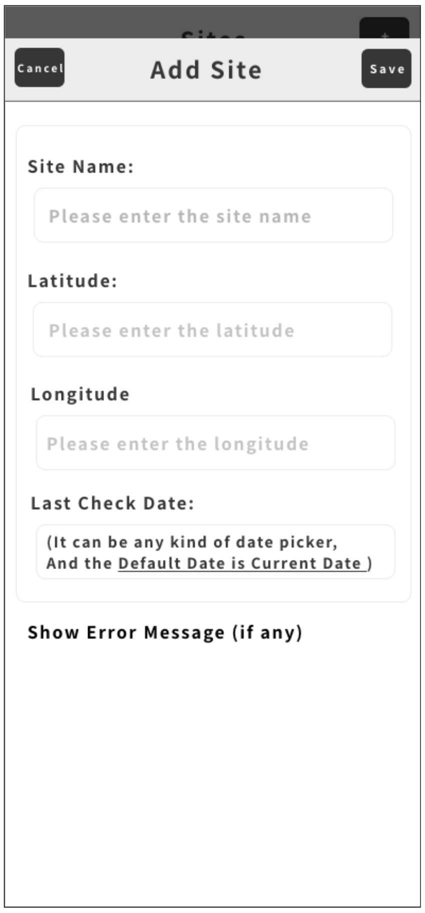
1. Navigation Bar [0.1]
   1.1 **"Report Issue"** title [0.1]
   1.2 **"Cancel"** and **"Save"** button [0.1]
2. Content Area [0.1]
   1.1 Site name label [0.1]
   1.2 Datetime label [0.1]
   1.3 **"Issue Type**:" & **"Description"** label [0.1]
   1.4 Issue Type Segmented control with options **"Type A", "Type B" and "Other"** [0.1]
   1.4 Description TextView [0.1]
   1.5 Error message label [0.1]



2.2 Functional requirements: [2.4]

1. The site name label shows **"Site Name: "** and the name of the selected site. [0.2]
2. The datetime label shows **"Datetime: " and the current datetime** (no need to update continuously) in yyyy-MM-dd HH:mm format. [0.3]
3. For the segmented control, the default selected item is **"Type A"**. [0.1]
4. When the user click **"Save"** button, the app will validate the data field (refer to step 5). If the all fields are validated, the app will store the issue and the page will be dismissed. Otherwise, the app will show **"Please input all the fields"** at the error message label but will not dismiss the page. [1.5]
5. Field validation requirements:
   a. the description text view should not be empty. [0.2]
6. When the user click "Cancel" button, the page will dismiss. [0.1]
7. **Design a proper data structure** to store the issues for display.

3. Add Site Page


Cancel
Add Site
Save
Site Name:
Please enter the site name
Latitude:
Please enter the latitude
Longitude
Please enter the longitude
Last Check Date:
(It can be any kind of date picker,
And the Default Date is Current Date)
Show Error Message (if any)

3.1 Elements include: [1.3]

1. Navigation Bar [0.1]
    1.1 **"Add Site"** title [0.1]
    1.2 **"Cancel"** and **"Save"** button [0.1]
2. Content Area [0.1]
    2.1 **"Site Name:"** label [0.1]
    2.2 Site name textfield with placeholder [0.1]
    2.3 **"Latitude:"** label [0.1]
    2.4 Lattitude textfield with placeholder [0.1]
    2.5 **"Longitude:"** label [0.1]
    2.6 Longitude textfield with placeholder [0.1]
    2.7 **"Last Check Date:"** label [0.1]
    2.8 A proper date picker in any form. [0.1]
    2.9 Error message label [0.1]

3.2 Functional requirements: [2.8]

1. The default last check date should be the current date. [0.2]
2. The date picker should show the selected date. [0.2] It also should allow picking a date without keyboard input. [0.2]
3. When the user click "Save" button, the app will validate the data field (refer to step 5). If the all fields are validated, the app will store the issue and the page will be dismissed. Otherwise, the app will show error message (refer to step 5) using the error message label but will not dismiss the page. [1.5]
4. Field validation requirements:
    a. the site name text view should not be empty. [0.2] (Error message: "missing field")
    b. the latitude text view should not be empty. [0.1] (Error message: "missing field")
    c. the input of latitude text view should be in numeric format. [0.1] (Error message: "invalid input")
    d. the longitude text view should not be empty. [0.1] (Error message: "missing field")
    e. the input of longitude text view should be in numeric format. [0.1] (Error message: "invalid input")
5. When the user click "Cancel" button, the page will dismiss. [0.1]

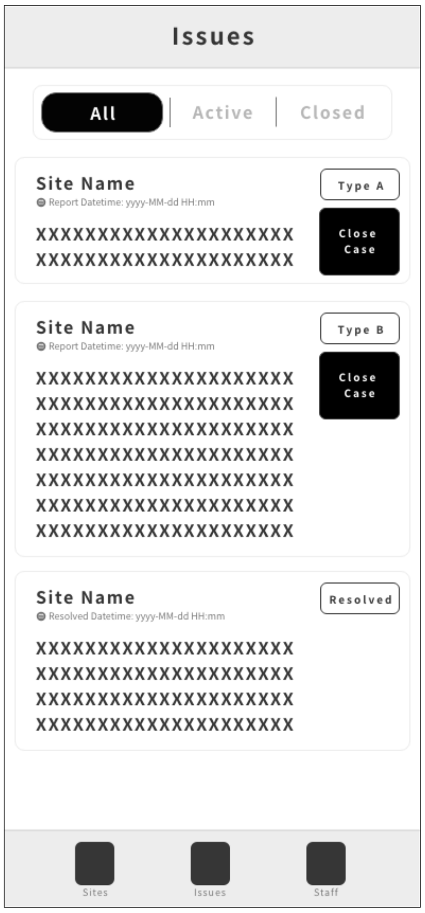
Test Coordinates for the GPS features:
```
Location of new sites
Aéroport de Lyon-Saint-Exupéry: 45.723760782731816, 5.084614166320578
Lyon Part-Dieu: 45.76065980303301, 4.861088477875932

Test Coordinate for new sites
Aéroport de Lyon-Saint-Exupéry: 45.72376828505583, 5.083991890428407
Lyon Part-Dieu: 45.76053220389201, 4.860975251020678
```

4. Issue Listing Page

4.1 Elements include: [0.9]

1. Navigation Bar [0.1]
    1.1 **"Issues"** title [0.1]
2. Issue Type Segmented control with options "Type A", "Type B" and "Other" [0.1]
3. Issues List [0.1]
    3.1 Issue cell [0.1]
    3.2 Site name label [0.1]
    3.3 Description text area[0.1]
    3.4 Issue Type/Resolved tag [0.1]
    3.5 **"Close Case"** button [0.1]



4.2 Functional requirements: [7.1]

1. Issue Listing:
    a. You should design and implement a UI element to indicate the issue list is empty. [0.2]
    b. The list should filter the issues according the segment control selection. [1.0]
    c. The default filter option is "All". [0.1]
    d. Sort the list in descending order by the report / resolved date time. [0.3]
    e. Issue list should show the data added by "Report Issue" Page [2.0] correctly.
    f. The cell should fit the size of the content of the description text view. [0.5]
    g. If the issue is not resolved, the list cell shows the type tag and the **"Close Case"** button, [0.5] and show the report date time according the the given format of the wireframe. [0.5]
    h. If the issue is resolved, the list cell shows the **"Resolved"** tag [0.5], and the whole the resolve date time according the the given format of the wireframe. [0.5]
    i. When the user click the **"Close Case"** of a particular issue, a confirmation dialogue must be prompted. [0.5]. Upon user's confirmation, the case should change to resolved status and store the resolved date time. And the UI of the particular cell should be update immediately. [0.5]

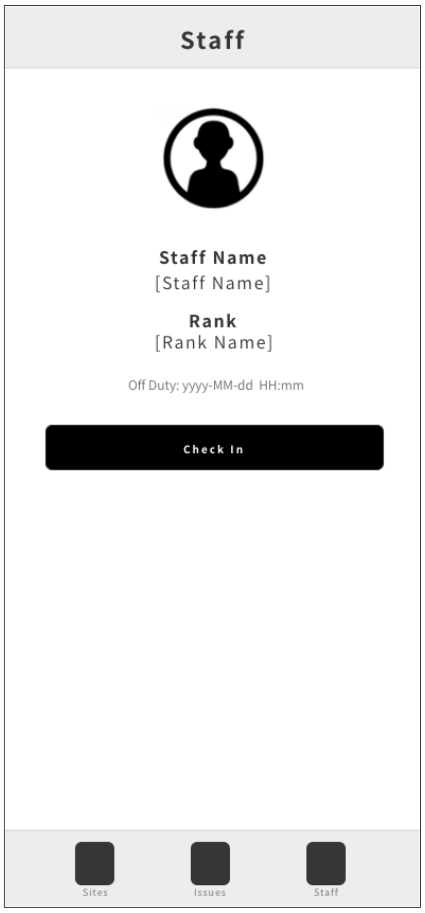
5. Staff Profile Page

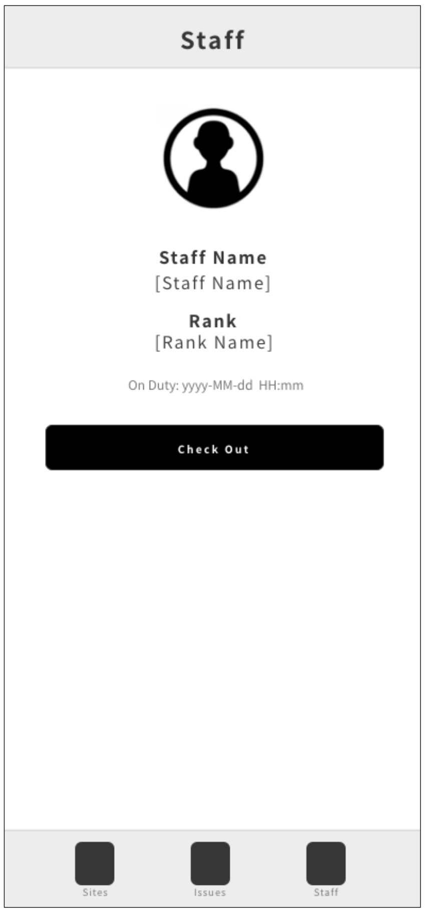
5.1 Elements include: [0.9]

1. Navigation Bar [0.1]
    1.1 **"Staff"** title [0.1]
2. Content Area [0.1]
    2.1 User icon [0.1]
    2.2 **"Staff Name"** label [0.1]
    2.3 Staff name label [0.1]
    2.4 **"Rank"** label [0.1]
    2.5 Rank name label [0.1]
    2.6 On/off duty label [0.1]

5.2 Functional requirements: [2.4]

1. The user icon show the given image. (media/icon.png) [0.1]
2. The staff name label / rank name label shows the data in the given data format. (data/staff.json) [0.2]
3. **Design the data structure to store the on/off duty status** with datetime. Then, according to the check-in/check-out status:
    a. Show **"Off Duty: " + checkout time** / **"On Duty: " + check-in time** properly. [1.5]
    b. Show the **"Check in" / "Check out"** button properly [0.5]
4. Click the **"Check In"** / **"Check Out"** button to show the QR Code Page. [0.1]





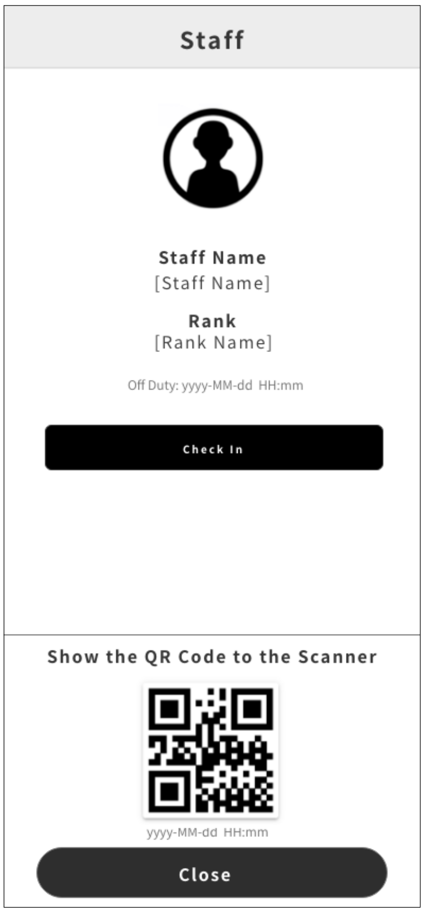
6. QR Code Page for Checkin / Checkout

6.1 Elements include: [0.4]

1. **"Show the QR Code to the Scanner"** label [0.1]
2. Image of QR code (media/qrcode.png) [0.1]
3. Current datetime label [0.1]
4. **"Close"** button [0.1]

6.2 Functional requirements: [0.7]

1. When the user clicks **"Close"**, the staff check-in/check-out status will be changed and stored accordingly, the page will be dismissed, and the change will be reflected on the staff page. [0.5]
2. The datetime label shows the current datetime (no need to update continuously) in yyyy-MM-dd HH:mm format. [0.2]



Instructions to the Competitor

1. You need to compress the dedicated files and related resource files into XX_Module_A.zip, export the mobile application as XX_Module_A.apk (for Android) or XX_Module_A.ipa/XX_Module_A.app (for iOS) executable files, and finally place them into the XX_Module_A folder.
2. The entire XX_Module_A folder should be pushed to the provided remote Git repository.

Note: XX refers to your country code.

Marking Scheme

Module A		
No.	Sub-scriterion	Mark
1	Inspection Site Listing Page	10.3
2	Report Issue Page	3.4
3	Add Site Page	4.1
4	Issues Listing Page	8.0
5	Staff Profile Page	3.3
6	QR Code Page for Checkin / Checkout	1.1
7	General Demands	6.1
8	Judgement	3.0
	Total	39.3

Judgement (3.0 marks):
The app is user friendly and well implemented. (1.5 marks)
0 - cannot run / no functionalities
1 - only a few functionalities are implemented
2 - most of the functionalities are implemented
3 - perfectly built, maybe with minor issues

The user interface and app navigation aligns with the given specification. (1.5 marks)
0 - many UI elements / pages are missing.
1 - the app somehow works but inappropriate UI / app navigation is implemented.
2 - most of the UI / app navigation works fine, but something is missing.
3 - perfectly built, maybe with minor issues