**CMPS 312 – Mobile Application Development**

**Midterm Practical Exam – 21 October 2021 (duration 2.5h)**

|  |  |
| --- | --- |
| Student Name (Student ID) |  |

**Objective:**Build Vaccine Management App on Android to demonstrate your Mobile development skills.

Sync cmps312-Lab repo then copy to your GitHub repo the midterm folder which contains an base solution that has all the needed dependencies and an *assets* folder that has *patient.json* , *vaccinestatus.json  and healthstatus.json* files.

In the drawable folder three images have been provided namely Infected, Healthy and Quarantine.

**Implementation Tasks**

Open the VaccineManagement App under midterm subfolder inside your GitHub repository.

1. **Patient List Screen**

|  |  |
| --- | --- |
| Graphical user interface, text, application  Description automatically generated | 1.1. Create a Patient List screen to display the patients. When the screen loads, get the details of patients and display them in a LazyColumn**.**  The image to show for each patient should be based on the HeathStatus, e.g., if the patient status is Healthy then the screen should Healthy.jpg  **Tip**: Use a patientList state variable in the PatientViewModel to hold the patients read from the provided PatientRepository.getPatients  1.2. Provide a delete icon next to each patient in the list to allow the user to click it to delete the patient.  **Tips**: - Use Icons.Outlined.*Delete*as the icon for delete  - Add deletePatient method to the PatientViewModel to delete the patient from the patientList  1.3. Add a Floating Action Button (FAB) to the PatientList screen to allow adding a patient. When the user clicks the add button, the app should navigate to the Patient Screen.  **Tip**: You may use a simple button instead of FAB if it is easier for you. |

1. **Add Patient Screen**

|  |  |
| --- | --- |
| Graphical user interface, application  Description automatically generated | 2.1. Create Patient Screen to allow the user to add a patient. The screen should allow the user to select the HealthStatus and the VaccinationStatus from a dropdown.  When the screen load, the HealthStatus and the VaccinationStatus dropdowns should be filled using the data returned from getHeathStatus and getVaccinationStatus provided by the PatientViewModel.  When the user submits the form, the app should add the new patient to the patients list and navigate back to the list of patients.  If the implementation is correctly done you should see the newly added patient displayed in the list of patients.    2.2. Add a title and a back arrow to the screen, when the back arrow is clicked then the app should navigate up to the Patient List screen.    **Tips:**  - Add getHeathStatus and getVaccinationStatus to the PatientViewModel to provide the data needed dropdown. These methods should call the provided methods in the repository.  - Add addPatient method to the PatientViewModel to add the new patient to the PatientList.  - Use PatientViewModel as a shared ViewModel to allow communication between the app screens. |

**Grading Rubrics**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **%** | **Functionality**\* | **Quality of the implementation** |
| 1. Patient List screen | 60 |  |  |
| 1. Add Patient Screen | 40 |  |  |
| Provide screenshots in *testing.docx* | - |  | *[-10pts if missing]* |

**\* Possible grading for functionality**: ***Complete and*** ***Working*** (get 70% of the assigned grade), ***Complete and*** ***Not*** ***working*** (lose 60% of assigned grade) and ***Not done*** get 0. The remaining grade is assigned to the quality of the implementation. Must submit screenshots in the **Testing-GradingSheet.docx**(otherwise -10pts).