

Introduction to OO Programming using Android

Continuous Assessment Instructions

SE-BSED (SE25)

Personal MediPal Application



©2017 NUS. The contents contained in this document may not be reproduced in any form or by any means, without the written permission of ISS, NUS other than for the purpose for which it has been supplied.

ASSIGNMENT

Introduction

To show your ability to create an object-oriented program, you will construct the Android application as detailed in this document.

Your work will be assessed not only on the basis of how well the program works, but also on how closely you have fulfilled the specification, how well you have structured your code and how well you have followed the Test Driven Development. You are advised to spend some quality time in team discussions to determine a suitable design.

Deliverables and Structure

You will deliver the following:

1. A ZIP file named **se25_XXX.zip** where **XXX** is your team number (e.g. P01, P02 etc. for Part-time teams, F01, F02 etc. for Full-time teams). The ZIP file will contain the complete project and folder structures of your Android application:

Note: You may use Android Studio IDE version 2.2 and above for development.

2. A Project Report named **report.doc**, placed in the **\docs** folder of the ZIP file. The report cover must clearly show the name of your team and a list of members. The report must contain:
 - UML class diagrams showing the domain classes (most important classes) of your software. The diagrams may be drawn using tools of your choice (e.g. WhiteStarUML, MS Word diagrams, scanned hand drawings, Enterprise Architect tool diagrams, etc.). They must show classes, their main attributes and methods (including key parameters), their relationships with roles and multiplicity, uses of inheritance, etc. No text explanation is required for the UML class diagrams.

Delivery and Support Mechanism

The assignment will be supported by our IVLE site (<http://ivle.nus.edu.sg/>). The course id is **SG4101** (Basic Software Engineering Discipline).

On the course site you will see that you have been grouped in your normal groups. The following facilities are available:

- The course Forum, which is used to discuss issues amongst yourselves and with the lecturers
- The course Workbin, where files are exchanged. In particular, the **Files (Workbin)** contains a folder for the peer assessments as well as two folders where **groups will upload their deliverables** (one folder for Part-time teams and one for Full-time teams)
- A Project space for each group, where the team can keep files for this project, and communicate with the facilities provided.

(Note: This is NOT the place for the final deliverables)

Upon completion of the project, and no later than the deadlines listed below, you will upload the completed assignment deliverables to the appropriate folder in the Workbin. Only one submission is required per team.

Peer Assessment

Each member of the team is required to privately submit a peer assessment form. Download the empty form from the Files (Workbin), rename it as your matriculation number, fill in your assessment and upload it in the same folder. Your submission is private – it can only be viewed by yourself and the lecturers.

Presentation Dates

Each team will be given **20 minutes** to present their design consisting of UML class diagrams (**Domain classes**). Instructors will give feedback on your design before you start on the testing and coding activities. The presentation schedule is as follows. Note: It is important to keep to your allocated time.

Venue: Meeting Room 4-05

Monday, 6 March 2017

Team	Time
1FT	9.30am to 9.50am
2 FT	9.55am to 10.15am
3 FT	10.20am to 10.40am
<i>Break</i>	<i>10.45am to 11.00am</i>
4 FT	11.00am to 11.20am
5 FT	11.25am to 11.45am
6 FT	11.50am to 12.10pm
<i>Lunch</i>	<i>12.10 to 1.15pm</i>
7 FT	1.15pm to 1.35pm
8 FT	1.40pm to 2.00pm
9 FT	2.10pm to 2.30pm

Saturday, 4 March 2017

Team	Time
1 PT	9.30am to 9.50am
2 PT	9.55am to 10.15am
3 PT	10.20am to 10.40am
4 PT	10.45am to 11.05am
<i>Break</i>	<i>11.05am to 11.25am</i>
5 PT	11.25am to 11.45am
6PT	11.50am to 12.10pm
7PT	12.15pm to 12.35pm
8PT	12.40pm to 1.00pm

Final Deliverable Deadlines

Part-time teams: Sat 25 Mar 2017

Full-time teams: Mon 27 Mar 2017

Software Restrictions

- You should use Android Studio ver. 2.2 and above.
- You should use Android SDK ver 19 and above.
- Use only Android Studio built in widget controls and layout.
- Use Java SDK 1.8.
- Use SQLite for manipulating data for local storage.
(Use of any remote database or other APIs such as XML etc. is **not** permitted)

How to score high...

1. Follow the specification carefully, make the software do what it is supposed to!
2. Ensure the application works smoothly. Fix the bugs!
3. Make sure it's installable: once you've built your ZIP file, try unpacking it on a different system, in a folder with a different name, with a JDK at a different location, and verify the batch files work properly.
4. If you think you know how to make this application more impressive, enhance it as appropriate.
5. Participate in the IVLE forums and help others with their questions.
6. Do not copy code. You will lose marks for direct use of code from solutions you find elsewhere, and for sharing solutions with other teams.
7. Write test case classes for as many important classes. Test important classes thoroughly.
8. Ensure that you create ample data to demonstrate your system

Assumptions

No system requirements can be specified completely and accurately, using only the English language ☺. Hence please make reasonable assumptions, where requirements are ambiguous. Ensure that you have documented them in your report. Clarify with the instructor on critical doubts.

SYSTEM REQUIREMENTS

You are assigned to implement an Android Personal MediPal Application for users to track and record their health statistics and manage medication.

The app will allow the user to input their personal and health bio, manage their medication, remind and notify them for medication, medical appointment. It will also remind the user when certain chronic medication are running low in stock so that the user could make the necessary arrangement to replenish the stock. The app will also allow the user to input and store In Case Of Emergency (ICE) contacts that will allow paramedics, passers-by or helpers to quickly identify you, treat you and contact the relevant authorities and next of kin immediately. The app is 100% Android-based. Data will be stored locally on the device and accessible by the application only, but not outside of the application using a structured data in a private database.

The design of your app should be:

- Modular and flexible enough to cater to new requirements and/or changes in the requirements.
- Take into consideration that most of the users might be elderly or people with chronic diseases.
- Proper domain objects should be factored into your design and implementation.

The app is designed primarily for only the user of the mobile device.

General Requirements

User Interface

The application will be everything the user can see and interact with. You should only use Android pre-built UI components, UI structured layout objects and widget controls to build your graphical user interface for your app. You should also use most of the other Android modules to provide dialogs, notifications and menus.

When the user starts the system for the first time, he/she will be directed through a list of help screens. You may have a settings options stored inside Shared Preferences to determine if this set of page screens will be displayed again on subsequent launching of the application. If this is the first time the user uses the application, he/she is required to input his/her personal and health bio in order to personalise the app for his/her usage. This may take the user through a sequence of screens to complete the necessary steps on storing the user's personal and health bio and/or the prescribed/self-medicated medication/supplements they are consuming.

Data Files

The data to be kept for the user of MediPal App, i.e. medication records, medical appointments, medication consumption and applied records, etc. should all be stored in an **SQLite** data file. The object data structure for each of the necessary information required is proposed in a later section.

The app should have the current state persisted at all times, so that even an anomalous termination of the application does not result in loss of data.

External Hardware (Optional)

No external hardware is required in this case. However, you may want to consider using the camera feature of the phone to scan and detect the end-user's personal bio, instead of keying into the app.

The end-user may also use this capability to scan and input medication prescription.

Data Requirements

Personal Bio

This table contains the personal information of the user. The following table structure defines what the table will store.

Attribute	Data Type/Length	Description
ID	Integer	Auto increment number that uniquely identify each row of record.
Name	Varchar(100)	Full name of the user as recorded or used by clinic and/or hospital.
DOB	Date	Date of Birth of the user. Age of the user can/may be derived from this attribute.
IDNo	Varchar(20)	Identity Number that uniquely identify the user.
Address	Varchar(100)	Store the Building Name, Location, Street Name, Level and Unit No of physical address of residence of the user.
PostalCode	Varchar(10)	Postal code of the physical address of residence of the user.
Height	Integer	Height of the user. This attribute, combined with the weight attribute in the Measurement table, can/will be used to derive the BMI of the user.
Blood Type	Varchar(10)	Blood Type of the user.

Health Bio

The medical history of the user should be kept in this table. This might include his/her medical conditions, allergies, etc. The following table structure defines what the table will store.

Attribute	Data Type/Length	Description
ID	Integer	Auto increment number that uniquely identify each row of record.
Condition	Varchar(255)	Medical conditions or allergies the user is suffering from.
StartDate	Date	Date the user was initially diagnosed with the medical condition.
ConditionType	Varchar(1)	Use the letter 'C' to identify condition the user is diagnosed with. Use the letter 'A' to identify allergy the user is sensitive to.

Categories

Medication are classified into categories (“supplement”, “chronic”, “incidental”, “complete course”, etc.). Each category is identified by a “category id”, which is a three-letter code (e.g. “SUP” for “supplement”). A category will exist in one or more medicine. You should be able to manage categories. Once the category is added, it can be edited but not deleted. The list below are proposed categories.

Category	Code	Reminder (Y/N/O)	Description
Supplement	SUP	N/O	User may opt to set reminder for consumption of supplement.
Chronic	CHR	Y	This is to categorise medication for long-term/life-time consumption for diseases, i.e. diabetes, hypertension, heart regulation, etc.
Incidental	INC	Y	For common cold, flu or symptoms happen to be unplanned or subordinate conjunction with something and prescription from general practitioners.
Complete Course	COM	Y	This may apply to medication like antibiotics for sinus infection, pneumonia, bronchitis, acne, strep throat, cellulitis, etc.
Self Apply	SEL	O/Y	To note down any self-prescribed or consume medication, i.e. applying band aids, balms, etc.

Y: Yes

N: No

O: Optional

The following table structure defines what the table will store.

Attribute	Data Type/Length	Description
ID	Integer	Auto increment number that uniquely identify each row of record.
Category	Varchar(50)	Short category name that uniquely identify the classification of the medicines.
Code	Varchar(5)	Three letter code that uniquely signify the category.
Description	Varchar(255)	Short description on the classification of medication for the category.
Remind	Boolean	Flag to determine whether user will be reminded to take medication for the medicine category.

Medicine

Medicine here refers to the medicines or supplements the user will be consuming. Each physical medicine in the store has a “medicine id” that identifies it uniquely. The medicine id is assigned automatically by the system. Each medicine will belong to a category. Some of the medicine attributes that should be captured includes medicine name, usage and dosage.

This table stores all medicine details, the user is prescribed with and/or consuming. The following table structure defines what you the table will store.

Attribute	Data Type/Length	Description
ID	Integer	Auto increment number that uniquely identify each row of record.
Medicine	Varchar(50)	Short category name that uniquely identify the classification of the medicines.
Description	Varchar(255)	Short description on the prescribed medicine for symptoms/medical diagnosis.
CatID	Integer	Reference to Category table for the classification of medicine.
ReminderID	Integer	Reference to Reminder table for the frequency of reminder daily for this medication.
Remind	Boolean	Turn on/off the reminder feature to notify user for consumption of medication.
Quantity	Integer	Quantity issued for the course of consumption.
Dosage	Integer	Refer to description below.
ConsumeQuality	Integer	Quantity to consume on on each frequency.
Threshold	Integer	Value that will trigger off a reminder to user to replenish the medicine for chronic diseases.
DateIssued	Date	Date the medicine was issued. This date should be used to compute the expiry of the medicine based on the attribute following.
ExpireFactor	Integer	Input value from 1 to 24. Value denotes the number of month(s) the issued medication will expire after the issued date.

Dosage can be in various units as proposed, i.e. pills, cc, ml, gr, mg, drops, pieces, puffs, units, teaspoon, tablespoon, patch, mcg, l, meq, spray.

Note: Some information may not be relevant for selected categories. (You may consider using enumeration feature to store this set of categories)

Measurements

The table structure will keep track of measurement information such as weight, blood pressure levels (systolic, diastolic, heart rate).

Attribute	Data Type/Length	Description
ID	Integer	Auto increment number that uniquely identify each row of record.
Systolic	Integer	Blood pressure systolic value of the user on measurement.
Diastolic	Integer	Blood pressure diastolic value of the user on measurement.
Pulse	Integer	Heart Rate of the user on measurement.
Temperature	Decimal(5,2)	Body Temperature of the user on measurement.
Weight	Integer	Weight of the user on measurement. This field will be used to compute the BMI of the user.

MeasuredOn	DateTime	Date and Time the measurement was taken
------------	----------	---

Consumption

This table structure will keep track of the type of medication that is consumed, the quantity consumed and when it is consumed.

The following table structure defines what the table will store.

Attribute	Data Type/Length	Description
ID	Integer	Auto increment number that uniquely identify each row of record.
MedicineID	Integer	Reference key to the consumed medicine in the medicine table.
Quantity	Integer	Quantity consumed based on the dosage specified.
ConsumedOn	DateTime	Date and Time the medicine was consumed.

Reminders

User will be notified of upcoming medication appointment(s), notification to consume medication.

The following table structure defines what the table will store.

Attribute	Data Type/Length	Description
ID	Integer	Auto increment number that uniquely identify each row of record.
Frequency	Integer	Number of times the user will be reminded based on the number of times the user need to take the medication per day.
StartTime	DateTime	Time the user will be first reminded for the day.
Interval	Integer	The number of hours before the next reminder will be carried out if the frequency is more than 1.

Appointment

Appointment information, such as the clinic of visit, consulting doctor, appointment date and time should be kept here.

The following table structure defines what you the table will store.

Attribute	Data Type/Length	Description
ID	Integer	Auto increment number that uniquely identify each row of record.
Location	Varchar(100)	Hospital and/or clinic of visit
Appointment	DateTime	Date and Time of the appointment
Description	Varchar(255)	Short description of any pre-requirements prior to appointment, i.e. fasting or abstinence requirements, etc.

ICE (In case of Emergency Contact)

This entity will keep track of the contact person, contact number (with possible dial capability), contact person's address, description or remarks. User should be able to categorise the ICE information to whether it is an emergency number, home physician (general practitioner), next of kin (NOK).

Note: Some information may not be relevant for selected categories.

Necessary validation should be performed on the contact number format.

The following table structure defines what the table will store.

Attribute	Data Type/Length	Description
ID	Integer	Auto increment number that uniquely identify each row of record.
Name	Varchar(100)	Name of contact person to contact
ContactNo	Varchar(20)	Contact Number of the contact person
ContactType	Integer	Type of contact person, i.e. NOK, GP, etc.
Description	Varchar(255)	Short description about the things to inform the contact person
Sequence	Integer	Priority of contact.

Functional Requirements

Manage Medication and Categories

Users should be able to manage the medication and the category of the medication they are consuming. This will include the dosage he/she is consuming.

Based on the selected category as described under Data Requirements, users may be able to set the frequency, number of times and time of reminder to consume the medication.

Frequency refers to the number of times the users will be reminded daily based on the number of times the user will take the medication per day. User will specify the initial time of reminder. Subsequent reminder of the day will be predetermined and computed.

Users should be able to input the number of pills collected for consumption and set the expiry date for the medication (where possible or necessary). Users should also be given the option to enable/disable and define possible replenish reminder when the pills drop below a certain level. Users should also be able to customise the time of reminder if the number of pills drops below the required threshold.

Manage Medical Appointment

Users should be able to add, amend or delete medical appointment. It should allow the users to keep track of the clinic/clinical department they will be visiting, the date and time of visit. Additional information such as the tests to be conducted, and pre-test requirements (e.g. fasting, etc.) maybe kept for user reference.

Users should be able to set reminders to remind them on the necessary pre-test requirements prior to medical appointment and also reminder for the medical appointment.

Manage Medicine Consumption

Users should be able to list his/her consumption history of medication. The app should also allow users to filter the list by medication and categories.

As a developer, you will need to design one or more views that will allow users or home physicians (general practitioner) to view the consumption history and whether any scheduled consumptions are missed.

Your design should also consider the ease of use in allowing user to indicate if they have taken the medicine. You may consider a simple listing (or report) that will indicate the type of medication they have taken by day, timing of consumption, and the medication they have missed. You might want to consider filtration of record history by year, month, week or day.

Manage ICE Contact (In case Emergency)

This feature will allow users to prioritise the preferred sequence of contact numbers in case of emergency. The contact can be a family member or friend. You may consider categorizing ICE into different categories as described under Data Requirements. Some proposed categories include Next of Kin, Doctors, Emergency Numbers.

The app should allow users to access the ICE feature for directly dialling the contact person's number based on the contact information provided. You may want to consider incorporating the ability to send SMS message to notify the ICE contact if he/she is uncontactable.

As per norm, you should allow users to amend the contact information and make necessary deletion of the information when it is no longer applicable.

Manage Measurements

The app should allow users to select the different types of health measurements they want to record. In this assignment, we will only be recording the following measurements.

Type	Measurement	Measurement Unit
Vital	Blood Pressure	
	- Systolic	mm Hg
	- Diastolic	mm Hg
	Pulse	bpm
	Temperature	Degree Celsius
	Weight	kg

Users should be able to capture these measurement information proposed for future reviews. Users should be able to make notes on each measurement.

Similar to the use case for Manage Medical Consumption, you should allow users to filter past records for viewing and reviewing by doctors and practitioners.

Reporting (Optional)

The system must be able to consolidate all the different records that are captured and output to the screen in some simple reporting format; i.e. listing of recorded values.

- ◆ Measurement for blood pressure
- ◆ Measurement for cholesterol level
- ◆ Measurement for weight
- ◆ Measurement for glucose level
- ◆ Consumption of medication

You may consider threshold indicators to highlight to users on whether measurement is within or outside of healthy/fit level.

These reports and past records may also be extracted and sent via email to respective doctors or family members for viewing and reviewing.

The above data/table structures and functionalities are not comprehensive and are purely designed for the purpose of this assignment. Please comply and follow strictly to the table structures provided. You are required to follow the table structure in the creation of your project. You are not required to provide or submit sample data. This will be carried out by the lecturer(s) involved when reviewing the application that you submit.

Once again, if you have any queries on the requirements, do clarify with the respective lecturer(s).

👍😊👍 *All the Best...*

*Hope you enjoyed the journey of Object-Oriented Android
Design and Development....👍😊👍*