

## How to Use this Template

1. Make a copy [ File → Make a copy... ]
2. Rename this file: **“Capstone\_Stage1”**
3. Replace the text in green

## Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [ File → Download as PDF ]
  2. Create a new GitHub repo for the capstone. Name it **“Capstone Project”**
  3. Add this document to your repo. Make sure it’s named **“Capstone\_Stage1.pdf”**
- 

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you’ll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

**GitHub Username:** hankarun

# Patient's Medical History Survey

## Description

Patient Survey application is a digital front end for the survey and historical patient data asking when you go to doctor. With this app doctors can create survey and keep record of the patients. Patient Medical History Surveys can be converted to pdf and be printed and shared.

Patient's medical history and information kept in the device for security. Doctors and patients can use same app, admin features like changing survey questions and patient history done via user defined password.

## Intended User

This application is intended for doctors and dentists, who wants to make medical history survey to their patients.

## Features

Users can enter their personal data,

Users can take survey,

Admin user can show historical patient surveys,

Admin user can change the questions,

## User Interface Mocks

### Screen 1

Brand Name

Welcome

This is a survey to  
get inf- of your med. history

Start

This is the welcome screen. When a patient opens the app this screen will help to start the survey. In the activity menu there is one option to enter the admin features. When you click it will ask for admin password. So only option for patient is start button.

### Screen 2

Patient Details

Name

Surname

Birth Date

Telephone

When patient press start screen there will be a dialog shown to the user. Dialog will ask user if he/she is previous user or not. If he/she is a previous user then by entering name, surname and birth date other details will fill automatically.

This screen will not pass to the other without filling required information. This is an activity with viewpager so pressing button or swiping changes the screen.

### Screen 3

A hand-drawn diagram of a survey screen. It features a central container with a title 'Question Group Name'. Below the title are three question cards. The first card is labeled '1' and contains the text 'Question Text' followed by radio buttons for 'Yes' and 'No'. The second card is labeled '2' and the third is labeled '3'. At the bottom left and right corners of the container are circular buttons with left and right arrow icons, respectively.

This is a fragment shown after user detail screen. All of the groups of questions in the survey shown user by the group order and a screen per group.

Arrows are using to move around different question screens.

Questions shown as cardview in a listview. Patient can select yes or no depending on the questions. No is pre-selected.

Screen 4

A hand-drawn diagram of an end screen. It features a central container with the text 'Congrats you finished.' underlined. Below this text is a rectangular button labeled 'End'. At the bottom left corner of the container is a circular button with a left arrow icon.

After all the questions finished. End screen is shown to the user. When it press the end button. Application saves the data and exits.

Screen 5

This is the doctor (admin) side of the application. User can reach this screen via password entered from greeting screen. This is a two fragment portrait layout. Doctor can select previous patient surveys, question groups, questions and settings. With search button user can search patients by name.

Screen 6

When a patient is selected. User can see their previous surveys and print or send them as email. Also user can change patient's details via a dialog.

Screen 7

This is a user edit dialog. Consists several edit text boxes to edit database data.

Screen 8

		+
Patients	<div>1. Group Text <span>△</span> <span>▽</span></div> <div>2. Group Text <span>△</span> <span>▽</span></div> <div>3. Group Text <span>△</span> <span>▽</span></div>	
Groups		
Questions		
Settings		

This is question group screen. User can see the groups, arrange them, add or remove them. To add user can use + button.

To delete user must click group and delete it from that dialog.

All info shown as cardview in a recyclerview.

### Screen 9

Group Edit

Group Text

---

Group Detail

---

This is question group edit dialog. When user clicks on a cardview this dialog will shown. There is also a delete icon in the dialog.

### Screen 10

		+
Patients	Group Name <span>▽</span>	
Groups		
Questions	1. Question <span>△</span> <span>▽</span>	
Settings	2. Question <span>△</span> <span>▽</span>	

Question listing window. Group names will be shown as a drop box at the top, depending on selected group, questions will be shown, arrange, added.

## Screen 11

When user clicked on a question question edit dialog will appear. User can change informations about questions. To delete a questions there will be a delete image button.

## Key Considerations

### How will your app handle data persistence?

Because this is a survey application, all data will be handled with SQLite database. A data provider will be responsible with database communication. Questions table will be prepopulated with the deployment of apk. Other information such as patient details, patient answers will be kept in another SQLite database.

Data provider class will be very complicated by pulling and writing data from two different database files, 4 different database tables. Different URI configurations will be needed.

### Describe any corner cases in the UX.

The users who purchase this application will be doctors but the real users will be patients. That's why the patient's side of the application must be informative.

When a patient starts to answer questions, arrows will move the viewpager but also swipe will be available.

When a user presses a back button it will be handled by the app and the swipe pager will be activated not the previous activity.

After survey started there is no way to return to the greeting screen, only by closing the app.

Describe any libraries you'll be using and share your reasoning for including them.

For example, Picasso or Glide to handle the loading and caching of images.

None different from base android libraries.

All UI will be designed with android material design rules using support libraries.

Coordination layout will be used.

## Next Steps: Required Tasks

### Task 1:

Create pre populated questions database in computer.

Prepare sqLitehelper for questions database.

Write content provider and determine uri types.

Test content provider show data with a listview for all uri types.

### Task 2:

Create main activity for greeting the user.

Create animations for the layout.

Create questions activity layout.

Prepare fragmentpager and populate it with user detail fragment and questions fragment.

Create question cardview.

### Task 3:

Finish model buildings for dataprovider.

Write data persistence inside the questions activity.

Test data across fragments.

Fill question fragments with cardview.

Implement cardview onclick actions and data storage.

### Task 4:

Implement end activity.

Show a summary of the survey.

Create models for user details and users answer.

Save survey result to the appropriate models.

### Task 5:

Create admin activity with two panel layout.

Populate fragments with user data, questions.

Prepare patient history dialog box.

Prepare question edit dialog box.

Prepare group edit dialog box.



Task 6:

Implement password protected access to the admin activity via menu button in main activity.

Test activities with dummy data.