2019

University of North Carolina Wilmington

Master of Science in

Computer Science and Information Systems

Proceedings

https://csbapp.uncw.edu/mscsis

ANDROID MOBILE APPLICATIONS DEVELOPMENT FOR HEALTHCARE DATA COLLECTION

Swanand Nalawade

A Capstone Project Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
Department of Information Systems and Operations Management

University of North Carolina Wilmington

2019

Approved by

Advisory Committee

George Schell, Committee Member 1 Devon Simmonds, Committee Member 2

______________________________ ______________________________

____
Elizabeth Baker, Chair

Accepted By

____
Dean, Graduate School

TABLE OF CONTENTS

ABSTRACT

Android Mobile Applications Development for Healthcare Data Collection. Nalawade, Swanand, 2019. Capstone Paper, University of North Carolina Wilmington.

The applications of smartphone technology are widespread but have left the medical industry largely unexplored. Doctors now have modern technology to diagnose and cure patients, but the collection of data and its subsequent storage is still antiquated. In some rural areas around the world, paper-based record systems are used to keep patient data. Even in adequately resourceful places, patients must make an appointment to see a doctor who can then tell them about their current condition. This system of patient care is highly resource intensive and presents recurring difficulty for long-term patients, especially those individuals suffering from cancer. Cancer patients need to show up regularly to report their current condition and any notable incidents of pain or distress to their oncologists. The motivation for this paper was to test if we could reduce the resources wasted on this operation, so that the process can be more efficient and would increase the quality of life of already troubled patients. In order to address the problem of unnecessarily difficult doctors' visits, one could develop an android application called PainSmart that would collect data directly from the users and send it to the doctor without having to visit the clinic in person. The data gathered by PainSmart will be locally stored on the device the patient is using and can be adapted to be saved in the doctor's local or centralized database. It was tested by the cancer patients under supervision of Dr. Jeeyae. Five patients were able to explore the pain app after only 20 minutes of training. (Choi, J., Baker, E., Nalawade, S., Peacock, A., Lee, H., Choi, W., 2018)

Keywords: healthcare, android, junit, data collection

LIST OF FIGURES

Figure Page

CHAPTER 1: INTRODUCTION

Healthcare is one of the most important industries in United States of America and all over the world. It has developed from traditional medicine to modern technologically savvy treatments. Along with the development of treatments, the IT infrastructure supporting this has also developed. Despite these improvements, the waiting time for the patients to see a doctor in USA is 18.5 days as of 2017. It typically takes about 11 minutes in the specialty of radiation oncology for the first-time emergency appointment but can be much longer for a regular checkup (Rege, 2017*)*. In rural areas there are much bigger problems to address in the field of healthcare, with paper-based data storage system being one of the biggest problems in need of. These systems are fragile, can be destroyed by disasters, accidents or insects and also consume a great deal of physical storage space. Advancements in paper-based data storage systems were made to store data on computers locally, which drastically reduced the storage space requirements. Despite this, the threat of data loss due to disasters still existed. Then with network connection infrastructure came cloud computing where all the data was to be stored in the cloud (Sencer, 2015). This is being used as of now, and the information which is stored is inputted by doctors or the staff and can be used for future reference. However, it is still a time-consuming routine (Free, Phillips, Watson, Galli, Felix, Edwards, Patel and Haines, 2013). The major other problem in this scenario is the loss of information due to forgetfulness of human nature. By the time a patient receives an appointment and gets to visit the doctor to report their past experiences, there is a high chance that the pain episode details might have faded from memory. This eventually leads to less accurate data and thus, low efficiency in treatment. To maintain data accuracy, the patients can keep records of their symptoms and experiences in a diary to show those records to the doctor next time they visit for an appointment. This can give accurate data, but it still requires a visit of patient to the doctor which is a waste of resources and can be painful for the patient. Additionally, the data is not being stored automatically, as the staff must enter the data manually from the patient visit records. It seems that there is a lot of room for improvement of the patient experience when managing their cancer pain, primarily making it easier for ill patients to communicate with their doctor about their pain so that they may get better recommendations on how to manage their pain. Yet it is a problem that can be solved by mobile applications used in healthcare (Boulos, Wheeler, Tavares, and Jones, 2011).

The application being developed from this paper in this capstone project, PainSmart, will take care of all a fore mentioned problems and will provide a functionality for the patients to send a pain episodes information to the doctors through an email. Due to PainSmart being a test for concept, the focus will be on data collection, local storage and sending it over email. The process of sending and storage of data could be modified to be used with current compatible cloud storage systems. The second section of this capstone will give a background of the current problem, while the third section will present PainSmart in depth as a solution for the need to provide enhanced tools to cancer patients. Chapter 4 will present the unit testing results performed on PainSmart, followed by a section describing PainSmart usage results and conclusions.

CHAPTER 2: BACKGROUND

Cancer is at one of the most dreaded diseases in the world, which has severe symptoms such as pain, fatigue, weakness and many others. The typical treatment for cancer lasts 6 to 9 months and is a very painful process. Since the advent of general anesthesia which opened doors for modern cancer surgery in 1846, there have been quite a few advancements in the attempt to make life easier for cancer patients. In 1903, after first use of radiation to treat cancer, the field of cancer treatment revolutionized. National Cancer Institute [NCI] was formed later to support the research on causes, diagnosis and treatment of cancer. In 1958, NCI scientists demonstrated combination chemotherapy to cure leukemia. After extensive research and support from society, adjuvant chemotherapy was demonstrated to cure early stage breast cancer. With many advances over time, the first gene therapy to be approved by FDA in 2017 became the latest mark in the history of this war (American Society of Clinical Oncology, 2019). However, even after all these achievements, we consider the treatment of cancer to be a painful one.

Cancer patients suffer through episodes of pain which can vary from minor pain to major breakthrough pain. These episodes of pain are important for evaluating the patient condition, and their pain levels which can help by determining and evaluating the right treatment. Sometimes in case of pain, the doctor may not be available at the immediate notice, but a cellphone usually is. Today, many people have access to cellphone devices, yet the medicine industry is lagging in having applications for collecting health related data. The only few available are for self-monitoring health and individual physiques. The applications when paired with monitoring devices like Fitbit, can be used to track the heart-rate of a user. There have been a few applications made to track chronic pain including the Chronic Pain Tracker, medications tracking using My

Pain Diary and Symptom Tracker, and paining body-parts using GeoPain (Erin Migdol, 2019). Surveys shows that few people use fitness apps in their day to day life, but many people do want to use those applications. Around 26% of people use the applications either regularly or occasionally. 15% of people refuse to use it, but a solid 59% of the target crowd claimed that they would like to use these kinds of applications (Statista Survey, 2017). This survey provides a solid idea that people are willing to switch to the mobile health applications rather than traditional applications. There is not, however, any application available for patients to directly send information to doctors as a replacement for follow-up process. Considering these things, we create an application which can take input from users or patients and store the data of an episode to send it to the doctor. This can save the patient a lot of time and effort for a trip to hospital to log data about the incident, and in turn would speed up the process of data collection. Capturing data as it occurs, or at least soon after occurring, would increase the accuracy of pain data.

PainSmart aims at tackling this problem to make the treatment process easier for patients in those 6 – 9 months by reducing the effort of the patients to carry out pain management in the treatment process. A major focus of the application will be based on making the app simple to understand and use. It will have a mainstream flow which would require less effort from the patient in pain. The usage of this application will give the head start in researching and developing more sophisticated applications in healthcare. Furthermore, a single database connecting all the patients, diseases, doctors, treatments and their analysis can prove to be the next change in the way we look at the medical industry. Android will be used for the operating system because of its popularity over other operating systems (Statista, 2018). PainSmart can help keep track of the past records, store new incidents, and send all the data to the respective doctor. PainSmart will provide a proof of concept about using smartphone technology in monitoring health rather than just fitness. The initial build will have the entry of data from users, but the potential expansion of this project can be tremendous, to include automatic health monitors. The project will allow patients to substitute the ongoing repetitive process of going to doctors for answering the follow-up questions and thus save time and resources of the doctor and the patient, while receiving better information on how to treat the patient's pain episodes.

CHAPTER 3: SOFTWARE DEVELOPMENT LIFE CYCLE

Dr. Jeeyae required an application for testing the usability and functionality of smartphone applications in the field of cancer treatment. The requirements were broken down into 3 parts. First being allowing the patient to register and collect his data locally on the device. Second being the patient can enter data of a pain episode and store it locally. Third was to enable the patient to send the data to the doctor over an email.

In android development, it is easy to create pages based on a story. Since we had the Unified Modeling Language (UML) diagram created, it was easy to divide the development into smaller targets. The work was divided into 3 phases based on the functionality requirement on completion of which, the application was supposed to be handed out to the patients for testing by Dr. Jeeyae.

Phase 1: History Data Collection

Phase One was the initial phase. With no or little prior experience, and the knowledge from online tutorials, we decided to use android studio as development tool using java for development language. The views were generated in Extensible Markup Language (XML) and called respective java pages for support. The history phase consisted of an initial eight pages out of which the first page was split into two at a later time. The result produced nine pages with an additional testing page to view collected data and beginning start and intro page. Basic selectors and form fields were used to collect data like text box, radio buttons, checkboxes spinners and dropdown lists. Page activity_history_a1 and activity_history_a1_1 focuses on getting primary information of the user.

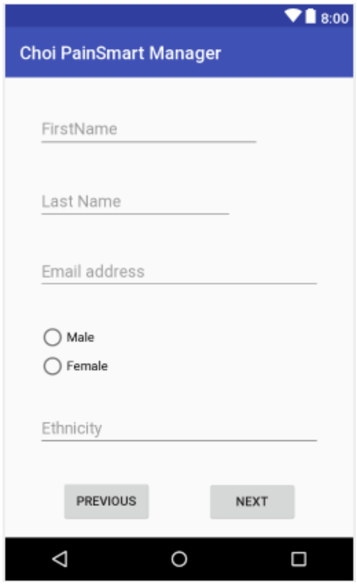
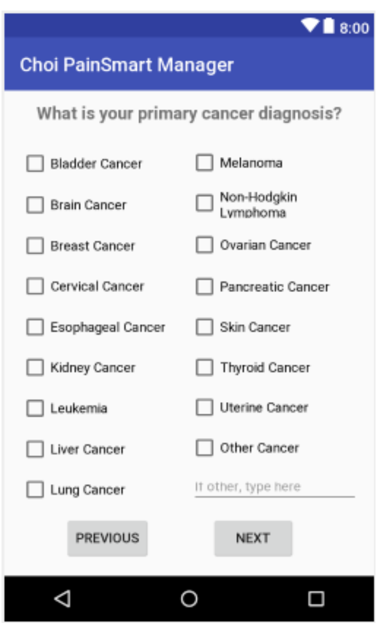
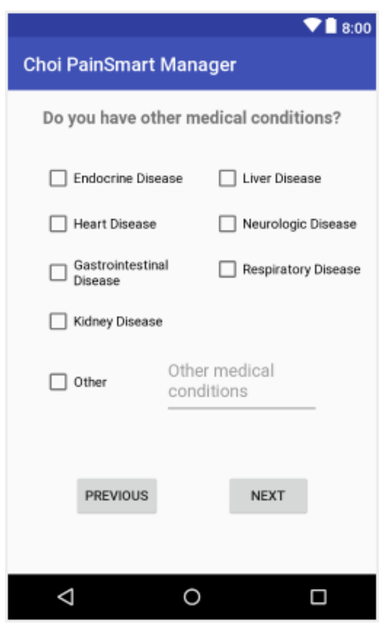


Figure 1: History a1

Figure 2: History a1_1

While from activity_history_a2 to activity_history_a7 focuses on gathering information about patient history and cancer detected. It also focuses on getting drug and abuse history.





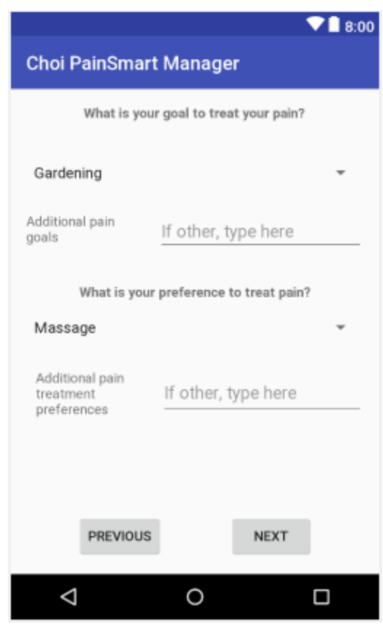


Figure 3: History a2

Figure 4: History a3

Figure 5: History a4

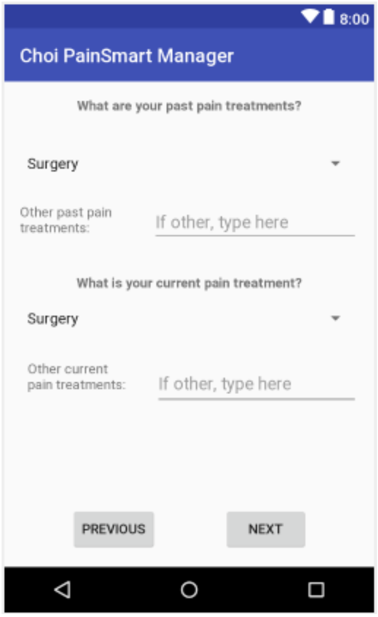


Figure 6: History a5

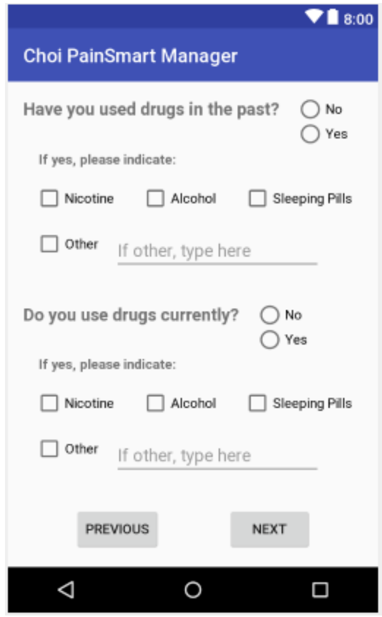


Figure 7: History a6

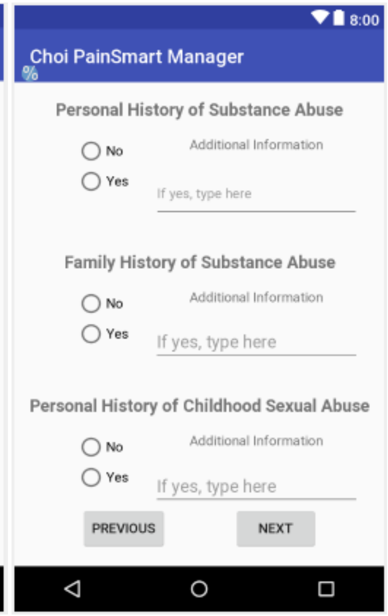


Figure 8: History a7

Activity_history_a8 is a confirmation screen that everything is submitted and has a redirection to the main screen.

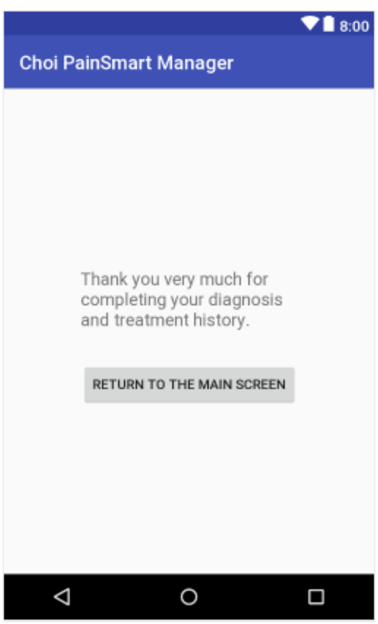


Figure 9: History a8

Phase 2: Episode Data Collection

By the time we were on phase two, we had gained insight into development and using Android Studio. The story was plotted for the flow of pages and multiple XML diagrams were prepared. Episode_A1 to Episode_A15 were the pages required to collect this data.

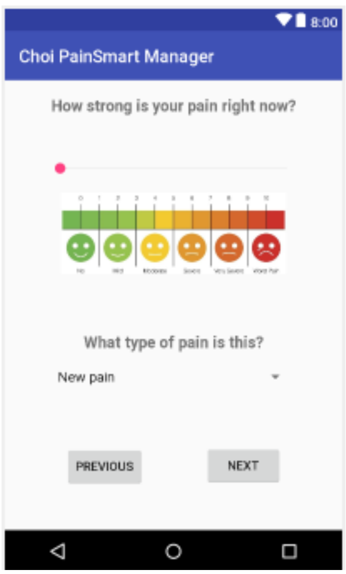


Figure 10: Episode a1

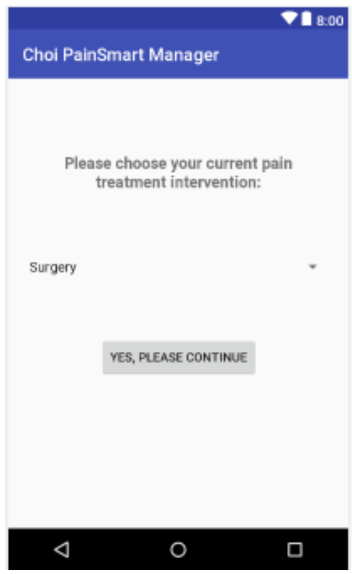


Figure 11: Episode a2

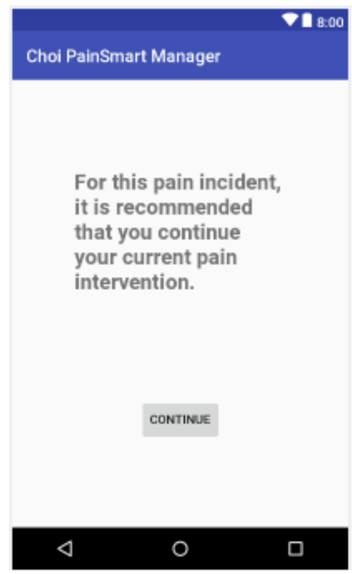


Figure 12: Episode a3

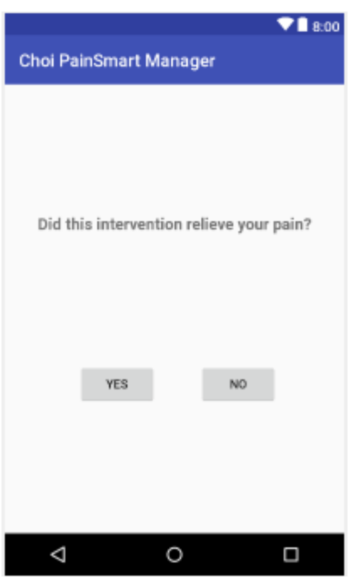


Figure 13: Episode a4

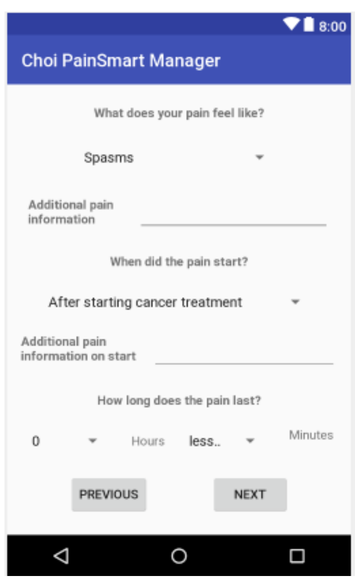


Figure 14: Episode a5

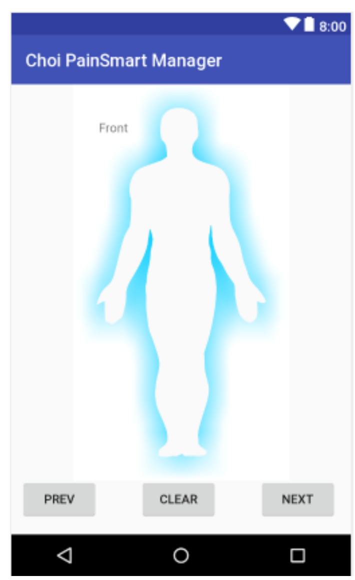


Figure 15: Episode a6

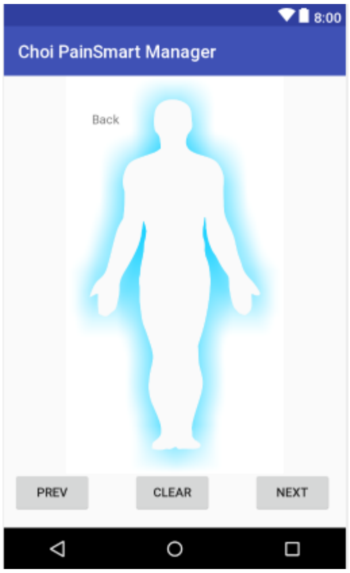


Figure 16: Episode a6__1

Figure 17: Episode a7

Figure 18: episode a8

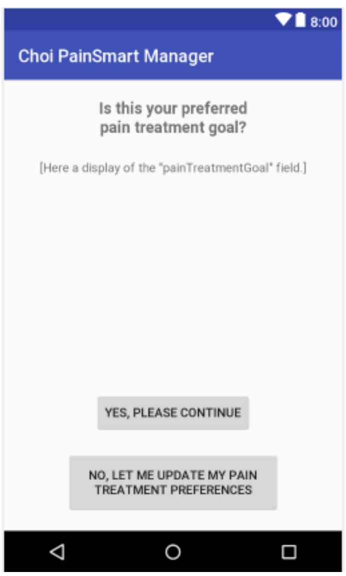


Figure 19: Episode a9

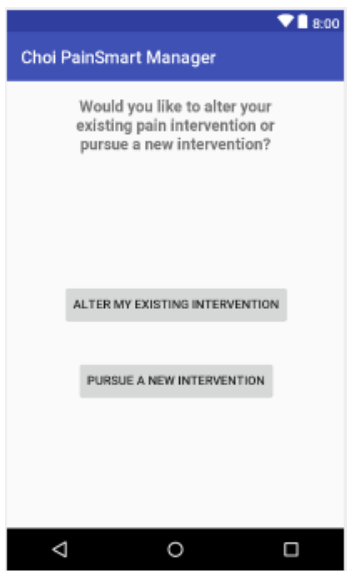


Figure 20: Episode a10

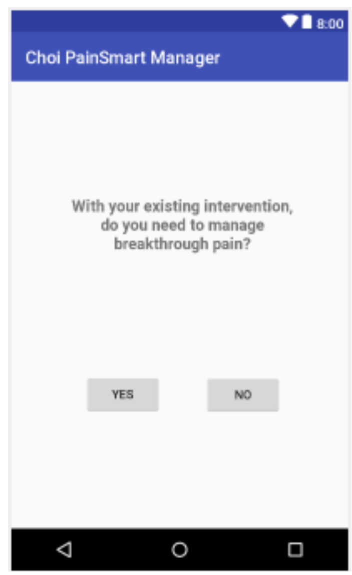


Figure 21: Episode a11

Figure 22: Episode a12

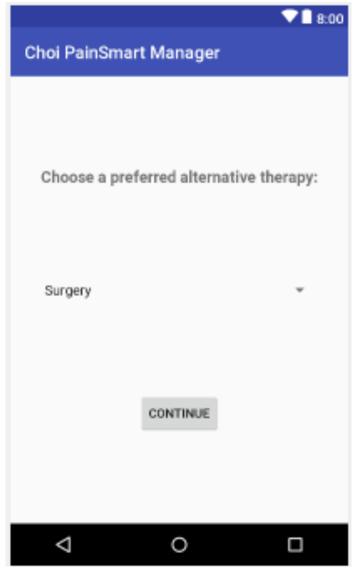


Figure 23: Episode a13

Figure 24: Episode a14

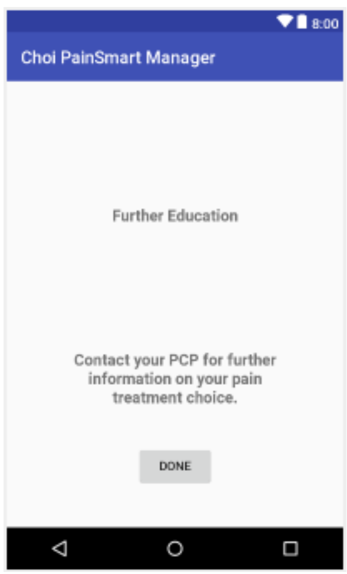


Figure 25: Episode a15

Episode_A6 and Episode_A6__1 were the ones kept aside for being different, and rest were prepared. Each spinner used in the pages had an associated array stored in Strings file in res folder. The image used for the pain scale was stored in drawable folder under res folder. For the design, A6 and A6__1, hollow images were used. The buttons placed under the image were initially transparent. They were placed in position to cover the background of hollow part of the image. The elevation used for buttons was less than elevation used for the image which gave those buttons a perception to be behind the image.

Phase 3: Database Formation

The initial storage used was a collection of static variables. They would retain the value over multiple sessions of the application. This data is deleted if the application is uninstalled. Later implementation used SQLite for creating a database. Two different databases would be required to store the data and each database will contain one table each. Database 1 has a table for history storage while Database 2 has a table for episode storage. Simple tables were implemented at 3NF. At the end of respective process, the data stored in static variables will be written into respective tables. Data stored can be converted to single string and viewed in a text format as shown in Figure 25 as per the incidents. It can be sent to the doctor via Email as seen in Figure 25.

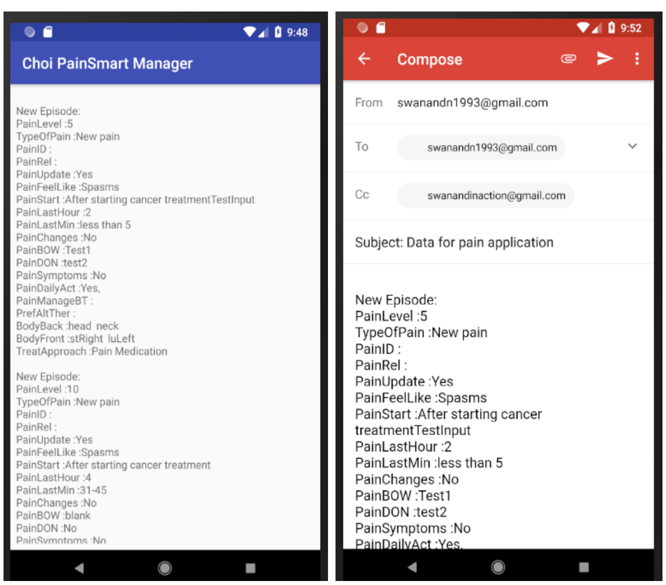


Figure 26: Phase 3 of Android UML designs.

CHAPTER 4: TESTING

To have a check of functionality, testing is a critical part. After completion of Phase 3, the testing environment was to be setup. Junit is a unit testing framework for Java programming language. Junit testing was used for this project and I intended to run multiple testcases for the inputs. The inputs from the users were tested for the correctness. Error checked was for right type of input and null values. The crude application currently developed will undergo series of changes as the multiple failed cases are resolved over time and if asked by Dr. Jeeyae as a requirement for her further research. Android studio makes it possible to implement unit testing that runs on local Java Virtual Machine (JVM) or an instrumented test that runs on a device (Android Developers, 2017). Using Espresso was a good choice for implementing testing (Valletta, 2017). Instrumented tests had the major focus and ran on hardware device- LG G7 ThinQ and emulator. These tests have access to Instrumentation Application Programming Interfaces (APIs) getting access to functional User Interface (UI) tests to automate user interaction. The automation included opening of an activity page, inserting texts in textboxes, inserting invalid inputs to field like a text in number field, selections on radio buttons and checkboxes, condition check based on selection of radio buttons, condition check based on selection of spinners and submitting the page.

The Java test files with respective test scenarios and their results:

1. HistoryA1Test - Checks with no Inputs.
2. HistoryA1Test2 - Checks with only radio button.
3. HistoryA1Test3 - Checks with names and wrong email(ssn6259).
4. HistoryA1Test4 - Checks with names and wrong email(ssn6259@uncw).
5. HistoryA1Test5 - Checks with no ethnicity.

6. HistoryA1Test6 - Checks with all right parameters.
7. HistoryA3Test - All right inputs.
8. HistoryA3Test2 - No text in others textbox when others checkbox is selected.
9. HistoryA3Test3 - Checkboxes other than others selected only.
10. HistoryA7Test - All correct inputs.
11. HistoryA7Test2 - Clicked yes but no text in description.
12. HistoryA7Test3 - Radio buttons not clicked at all.
13. EpisodeA1Test - Checks working of spinner, seek bar and activity load.
14. EpisodeA1Test2 - Checks spinner on EpisodeA1, goes to A2, checks load of A2, goes to A3 because of specific selection made on A1.
15. EpisodeA2Test - Checks flow of activity from EpisodeA2 to A5.

User testing was conducted by Dr. Choi with 5 cancer patients. Getting feedback from them for usability played an important role in finalizing the artifact.

CHAPTER 5: RESULTS

Overall, the project covered multiple aspects of the project development life cycle.

Following were the test results of previous mentioned cases which were used to test the crude application:

1. HistoryA1Test – Failed. Null check required.
2. HistoryA1Test2 – Failed. Null check required.
3. HistoryA1Test3 – Failed. Format check required.
4. HistoryA1Test4 – Failed. Format check required.
5. HistoryA1Test5 – Failed. Null check required.
6. HistoryA1Test6 – Passed.
7. HistoryA3Test – Passed.
8. HistoryA3Test2 – Failed. Null check required.
9. HistoryA3Test3 – Passed.
10. HistoryA7Test – Passed.
11. HistoryA7Test2 – Passed.
12. HistoryA7Test3 – Failed. Null check required.
13. EpisodeA1Test – Passed.
14. EpisodeA1Test2 – Passed.
15. EpisodeA2Test – Passed.

The UI testing score was 77.5. Score above 68 is considered an above average score on System Usability Scale (Usability.gov, 2018). The average age of patients was 59.2 years. Three out of five users asked details for recreational drugs related selections.

One user commented on body diagram being too big. One user commented that they found the body diagram to be anatomically correct. More clarity of questions is required with more pain goal options. Time spent on the application by users varied in the range of 7 minutes 17 seconds to 17 minutes 2 seconds. Mean time spent was 11 minutes 47 seconds. This was after 20 minutes of initial one-time training (Choi, J., Baker, E., Nalawade, S., Peacock, A., Lee, H., Choi, W., 2018).

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

PainSmart is a solution which provides a detailed pain assessment for cancer patients to proactively manage their condition. The application developed is still crude and failed a few null acceptances and format checkers but is enough to test the usability of the concept. Many advancements can be further done in this project to make the inputs refined. For example, addition of further clarity and explanation of questions can be done. Creation of a centralized database system for doctors to access in real-time would be a major step ahead. The users finding the application appealing and easy to use proves that this concept has a scope if exploited correctly. PainSmart is going to make life easier for cancer patients and in the process going to save lot of resources for the doctors.

Though there are a few applications on the market to collect health related data from users, there are barely any applications available to collect data from patients undergoing cancer treatment and there needs to be a lot more exploring in this area. Care providers would benefit from all collected data. The questions can be modified for further iterations to have more intrinsic data entry. PainSmart can serve as a prototype for many future health monitoring applications.

REFERENCES

American Society of Clinical Oncology. (2019). Cancer Progress Timeline. *ASCO: Connection.* Retrieved from https://www.asco.org/research-progress/cancer-progress-timeline

Android. (2018). [Table and a pie chart of android operating systems used recently April 16, 2018]. *Platform Versions.* Retrieved from https://developer.android.com/about/dashboards/index.html

Android Developers. (2017). Fundamentals of testing. *Android Developers*. Retrieved from https://developer.android.com/training/testing/fundamentals.html

Boulos, Wheeler, Tavares, & Jones. (2011). How smartphones are changing the face of mobile and participatory healthcare: an overview with example from eCAALYX. *Biomedical Engineering Online.* Retrieved from https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-10-24

Choi, J., Baker, E., Nalawade, S., Peacock, A., Lee, H., Choi, W. (2018). A Framework Facilitates Development of a Mobile App. *Studies in Health Technology and Informatics*, 250, 97-100.

Free, C., Phillips, G., Watson, L., Galli, L., Felix, L., Edwards, P., Patel, V., & Haines, A.. (2013). The effectiveness of mobile-health technologies to improve health care service delivery process: A systematic review and meta-analysis. *PLOS | Medicine.* Retrieved from http://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1001363

Migdol, E. (2019). Nine Apps that Can Help You Keep Better Track of Your Pain. *The Mighty.* Retrieved from https://themighty.com/2018/11/chronic-pain-best-tracking-apps/?utm_source=Autoimmune_Page&utm_medium=Facebook

Rege, A. (2017). Patient wait times in America: 9 Things to know. *Becker's Hosptial Review.* Retrieved from https://www.beckershospitalreview.com/hospital-physician-relationships/patient-wait-times-in-america-9-things-to-know.html

Sencer. (2015). Is storing healthcare data in the cloud a good idea? *KQED Education.* Retrieved from https://ww2.kqed.org/education/2015/11/18/is-storing-health-care-data-in-the-cloud-a-good-idea/

Statista Survey. (2017). Percentage of U.S. adults who would be willing to use an app to measure health metrics as of 2017. *Statista - The Statistics Portal*. Retrieved from https://www.statista.com/statistics/698610/us-adults-that-would-use-an-app-to-measure-health-metrics/

Statista. (2018). Global Market Share Held by Smartphone Operating Systems in Sales to End Users from First Quarter 2009 to Second Quarter 2018. *Statista – The Statistics Portal.* Retrieved from: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/

Valletta, N. (2017). Android UI Tests: Espresso Fundamentals. Pluralsight. Retrieved from https://www.pluralsight.com/courses/android-ui-tests-espresso-fundamentals?aid=7010a000001x3JXAAY&promo=&oid=&utm_source=non_branded&utm_medium=digital_paid_search_bing&utm_campaign=Bing_US_Dynamic&utm_content=&s_kwcid=AL!5668!10!77446913240513!dat-2329246664020045:loc-190&ef_id=WwVvwAAAC5CfwJJ_:20181213175426:s

Usability.gov (2019) System Usability Scale. Retrieved from https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html