

Test Results

Smart Health Consulting

Team Members:

- Mario Osborn
- Sebastian De La Cruz
- Meenakshi Karthikeya
- Jonathon Rice
- Luke Myers
- Austin Peace

Contents of this Document

Description of Test Environment

Results of Individual Test Cases

Conclusion

Description of Test Environment

Between the planned testing and the actual testing phase, there was only one difference: no testing occurred on physical devices. Although the planned testing specified performing a small amount of tests on a physical device, due to time constraints, no tests were able to be performed on physical devices. Although this in theory should not complicate pushing the software to physical devices, we were unable to conduct a couple of tests that relied on using a physical device rather than the Android Virtual Device (AVD).

Results of Individual Test Cases

Test results in **green** indicate successful tests, test cases in **red** indicate an unsuccessful test and include a brief description of why it did.

Test Cases for Registration

No.	Test Objective	Test Description	Expected Results	Test Results	Tester
R.1	Verify text	Check placement of	Objects on signup	Pass	Mario
	fields and buttons	objects on screen	page should be positioned correctly on screen		

R.2	Check blank information	Leave text fields empty	User should not be allowed to register and an error message should display	Pass	Jon
R.3	Check only required fields with acceptable data	Fill all required fields with acceptable data and click on signup button	Account should be created in the system with all required information attached and user should be brought to the login screen	Pass	Jon
R.4	Check optional fields with required fields	Fill all required and optional fields with acceptable data and click on signup button	Account should be created within the system with encrypted password and with all corresponding information asked at signup and user should be brought to the login screen	Pass	Jon
R.5	Check password with insufficient strength	Fill password field with a string that does not contain numbers, letters, and symbols Password: 1a2b3c	Proposed password should not be accepted by system and an error message should display	Pass	Jon
R.6	Check age field with non-acceptable age value	Input birthdate details that are under the age of 18 DOB: 1/1/2010	Account should not be created and an error message should be displayed	Pass	Jon
R.7	Check invalid username	Input username that already exists in database into select username field Username: user123	Account should not be created and an error message should display to screen	Pass	Jon

Test Cases for Login

No.	Test Objective	Test Description	Expected Results	Test Results	Tester
L.1	Verify text fields and buttons	Check placement of text fields and buttons on screen	Objects on login screen should be positioned correctly on screen	Pass	Mario
L.2	Check valid username with blank password	Input valid username with the password field left blank Username: user123	User should not be allowed to login and associated error message should display	Pass	Sebastian
L.3	Check valid username with invalid password	Input valid username with incorrect password Username: user123 Password: wrongpass1!	User should not be allowed to login and proper error message should display to the screen	Pass	Sebastian
L.4	Check valid username with correct password	Input valid username with correct password Username: user123 Password: correctpass1!	User should login to appropriate user page after system authenticates the user and access to the application is now allowed	Pass	Sebastian

Test Cases for Appointments

No.	Test Objective	Test Description	Expected Results	Test Results	Tester
A.1	Create a new service	While logged in as a doctor, navigate to the 'add service' page located under the profile page. Once there input values as instructed and press 'create'	Service should be created and stored on database linked to the specific doctor. It should be searchable from a patient's perspective	Pass	Austin

		Name: "Test Service" Date: 12/15/2017 Time: 8:00 AM Price: \$100			
A.2	Request an appointment	After following the steps described in M.1, click the first result and select 'request appointment'	The requested appointment should be added to the user's 'requested appointments' page and to the associated doctor's 'view pending appointment requests' page	Pass	Luke
A.3	View appointment details	While logged in as a doctor with at least 1 pending appointment approval, navigate to the 'view pending appointment requests' page under the profile page and click on a pending appointment	The details of the appointment (patient, doctor, service name, cost, date, and time) should be visible on a new page	Pass	Luke
A.4	Accept an appointment	Follow the steps detailed in A.3 and click 'accept'	A text field should appear where the doctor can add information. The appointment should show up on both the appropriate patient and doctor's calendar	Pass	Luke
A.5	Decline an appointment	Follow the steps detailed in A.3 and click 'decline'	A text field should appear where the doctor can provide a reason for declining the appointment request to the patient	Pass	Luke
A.6	Check conflicting appointments	While logged in as a patient with at least 1 appointment	An error message should display indicating that the	Fail The second	Meena

		already scheduled, search for an appointment that occurs any time during the already scheduled appointment and click 'request appointment'	user is not allowed to create conflicting appointments. No change should be made in the database	(conflicting) appointment is added to database and both appointments are added to patient's schedule	
A.7	Check appointment limit	Within the same 12 hour period, find 3 different services and request an appointment for all three	An error message should display after attempting to request the third appointment telling the user they are not allowed to request more than 2 appointments in a 12 hours period	Fail No error message is displayed and all appointments are added to patient's schedule	Sebastian

Test Cases for Calendar

No.	Test Objective	Test Description	Expected Results	Test Results	Tester
C.1	Verify appropriate calendar for user	Login as any user and visit the calendar page	The calendar with appointments tied to the specific user should be visible	Pass	Mario
C.2	Check calendar appointment for a specific date	Login as any user with at least one appointment, click on the calendar date for which that appointment exists	The appointment(s) for the user should for that specific date should display to the screen	Pass	Mario

Test Cases for Medical History

No.	Test Objective	Test Description	Expected Results	Test Results	Tester
MH.1	Upload medical history	Logged in as a patient at the “Medical History” portal, upload medical history Username: testPat Password: passwr1!	Created medical history should be uploaded to database and linked to the specific user’s account	Pass	Luke
MH.2	Add completed appointment to patient’s history	While logged in as a patient, complete an approved appointment with a doctor Username: testPat Password: passwr1!	Appointment details should be added to the patient’s appointment history	Pass	Meena
MH.3	View/download a patient’s medical history	While logged in as a doctor with an appointment with a patient that has allowed the doctor to view their medical history, go to patient’s medical history under their profile and click ‘download’	A PDF with the patient’s medical history should download and become viewable	Pass	Meena
MH.4	Check hidden medical history feature	While logged in as a doctor with an appointment with a patient that hasn’t allowed the doctor to view their medical history, click on the patient’s medical history under their profile	A page with the message “This user has not allowed you to view their medical history” should appear and the patient’s medical history should not be visible	Pass	Sebastian

Miscellaneous Test Cases

No.	Test Objective	Test Description	Expected Results	Test Results	Tester
M.1	Search for a service	While logged in as a patient, navigate to the 'Search for a Service' page located under the profile page. In the search bar, search for "dentist"	A list pulled from the database should show all results related to "dentist"	Pass	Luke
M.2	Search for a service with empty string	While logged in as a patient, leave the search bar then search	A pop-up should display stating that the search field was left blank	Fail This test case failed as during development, we decided to change the functionality of this test objective. We instead decided to have an empty string return all appointment listings.	Jon
M.3	Search for a service using all filters	Complete the steps described in M.1 and then click on 'filter' and select the following filters Speciality: "dentist" Price: <\$500 Date: 12/15/2017 Time: 2:00 PM	Results that fit the filters should be the only ones to display	Pass	Jon
M.4	Check messaging feature	Follow the steps outlined in A.4, then visit the associated patient's profile and select the 'message' button	The messaging page should display and allow the doctor to send messages to their patient	Pass	Sebastian

M.5	Verify received message	Follow the steps outlined in M.4, then type the string 'test' and select send	The message should only display in the patient's 'inbox' as well as the doctor's 'sent messages'	Pass	Sebastian
M.6	Check profile image	Logged in as any user, navigate to the profile page, select 'choose image' then select any image and select 'upload'	The selected image should now display as the user's profile image and be stored in the database	Pass	Mario
M.7	Verify password encryption	With at least 1 user registered in the system, go to the Firebase project linked to the software and check the list of all registered users	The list of all usernames should be displayed along with their encrypted passwords	Pass	Luke
M.8	Verify device compatibility	On a physical android phone running the required software and hardware, visit the application's page on the Google Play Store	A message should appear stating that the application is compatible with the device based on the manifest.xml document that accompanies the application	Fail As no testing was able to be conducted on physical devices, there was similarly no time to publish the application on the Google Play Store	Meena
M.9	Verify device incompatibility	On a physical android phone running SDK 22, visit the application's page on the Google Play Store	A message should appear stating that the application is not compatible with the device due to the software version. This is based on the manifest.xml document that accompanies the application	Fail See above explanation	Meena

Conclusion

Although not all of the test cases passed, the test cases that dealt with the required functionality of the application all passed. With this in mind, the software is ready to be delivered as it meets the stopping criteria defined in the test plan. Out of the five failed test cases, 2 failed due to a lack of using a physical device, 1 failed due to a functionality definition change for the test objective, and the last 2 failed due to lack of implementation. These last 2 are not required functionality but should be delivered in the next version of the application. Aside from that, no additional testing is required. The only concern for maintenance would be how the application interacts with Firebase, as Firebase has proven to us during development to have more outages than expected.

User Manual

Smart Health Consulting

Team Members:

- Mario Osborn
- Sebastian De La Cruz
- Meenakshi Karthikeya
- Jonathon Rice
- Luke Myers
- Austin Peace

Modification history:

Version	Date	Who	Comment
v0.1	10/31/17	Mario	Skeletal UI base completed
v0.2	11/12/17	Sebastian	Added Firebase login functionality, revamped UI, minor search implemented
v0.3	11/15/17	Mario	Added User class and settings
v0.32	11/20/17	Austin	Added Doctor Registration page
v0.35	11/22/17	Jon	Added image upload for profile picture
v0.4	11/22/17	Austin	Implemented Messaging functionality
v0.6	11/23/17	Jon	Doctors can now add appointments
v0.7	11/24/17	Luke	Added Appointment Search
v0.8	11/25/17	Meena	Added Calendar and PDF generation
v0.9	11/26/17	Austin	Added MR functionality
v0.95	11/27/17	Mario	Changed MR and UI changes
v1.0	11/27/17	Sebastian	Bug fixes, UI changes
v1.1	11/27/17	Austin	Optimized PDF generation

Contents of this Document

System Description

System States

System Functions

Appendix – Acronyms and Abbreviations

System Description

This product is an Android application that allows users to sign up as either patients or doctors. Patients are allowed to search and select from doctor-created appointments, chat with doctors via a real-time messaging system, view upcoming appointments, and have a medical history page. Doctors are allowed to create appointments, accept/decline appointment requests from patients, message patients, view medical histories of clients, and edit their profile.

This application makes is simple for patients to find and receive the care that they need at a predetermined and transparent price in a competitive marketplace. Through the use of this application, doctors will have another method to generate business and manage their patients through the use of the schedule within the application. Patients will be able to quickly sign up for appointments without having to make a phone call or go through the hassle of travelling to a physical location to make an appointment. These type of patients will get the most out of our software.

System States

Using the System (online): When using this system online appointments are updated real time via Google's Firebase. All functionality will be available.

Using the System (offline): When the system is offline limited functionality will be available. Appointment actions and messages will be sent to the local Firebase cache and will wait for reconnect before sending the information to other users.

Using the System as a Patient: When using the system as a patient the user can search appointments, sign up for appointments, view future appointments, as well as their personal medical record, and message doctors they signed up for.

Using the System as a Doctor: When using the system as a Doctor the user can view their listed appointments, view upcoming schedule, create new appointments, view patient medical history, and message patients.

System Functions

Create account (patient): Launch application (**Fig. 1**) → Select Sign Up → Enter email and create password (**Fig. 2**) → Select Sign Up → Select Sign Up as Patient → Fill out information (**Fig. 3**)



Fig. 1: Launch Page



Fig. 2: Sign-up Page



Fig. 3: Patient Registration

Create account (doctor): Launch application (**Fig. 1**) → Select Sign Up → Select Sign Up as Doctor → Fill out information

Create appointment: Login as a doctor → Navigate to the Offered Appointments tab → Select the floating plus button → Fill out appropriate information (**Fig. 4**)



Fig. 4: Create Appt. Page

Search appointments: Sign in as patient → Fill out search parameters (search page is the default page upon signing in) (**Fig. 5**) → Select Search (magnifying glass image) → Select an appointment



Fig. 5: Patient Home / Search Page

View future appointments: Sign in as a patient or a doctor → Navigate to My Appointments tab (**Fig. 6**)

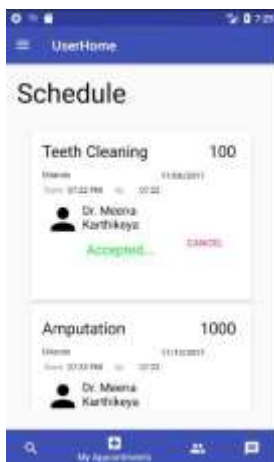


Fig. 6: My Appointments Page

Messaging: Sign in as a patient or a doctor → Navigate to the messaging tab (**Fig. 7**) → Select a conversation



Fig. 7: Messages Page

View patient medical history (doctor): Sign in as a doctor → Navigate to the patient-viewer tab (**Fig. 8**) → Select a patient → View or update medical history



Fig. 8: My Patients Page

View patient medical history (patient): Sign in as a patient → Select the menu button on the top left (**Fig. 9**)→ Select the Medical History button (**Fig. 10**) → Select Generate PDF (**Fig. 10**) → Navigate to Files application on device → Select generated PDF (**Fig. 11**)



Fig. 9: Patient Menu

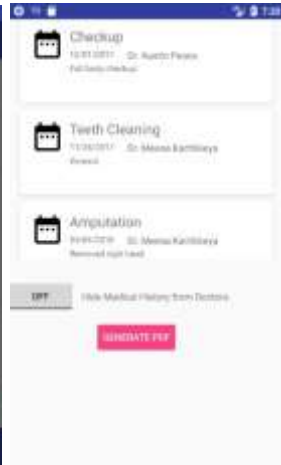


Fig. 10: Medical History Page



Fig. 11: Medical History PDF

Appendix – Acronyms and Abbreviations

Build Instructions

Smart Health Consulting

Team Members:

- Mario Osborn
 - Sebastian De La Cruz
 - Meenakshi Karthikeya
 - Jonathon Rice
 - Luke Myers
 - Austin Peace
-

Materials Required

In order to install and run the system, the user will require either an Android Virtual Device (AVD) with at least the Android 8.0 SDK or a physical Android device (phone) that is compatible with Android 8.0. The host computer will need at least 8GBs of RAM. The user will also need the Doctr™ source code.

To build the source code and run on a Android Virtual Device the user must utilize: Android Studio 3.0, the Android 8.0 SDK (API 26), an AVD with Android Oreo (Android 8.0) with Google Play services, and an internet connection.

To install and run on physical hardware the user will need Android Studio 3.0, the Android 8 SDK (API 26) and an Android Phone meeting the specific hardware requirements outlined in the Android 8.0 Compatibility Definition, which can be found at <https://source.android.com/compatibility/8.0/android-8.0-cdd.pdf>

Build Instructions

To build in Android Studio select build on the ribbon and then select Build APK(s). The APK file will be located under your project directory in app/builds/outputs/apk/debug. Move the file to the android phone using a usb and open it on your phone to install the application.