Test Plan

Smart Health Consulting

Team Members:

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

Contents of this Document

Overall Objective for Software Test Activity

We expect the software test effort to identify any and all possible failure states with information describing how to reach each individual failure state. We also expect to verify and validate the software against the design descriptions set by the developers and the specifications set by the client. We hope to ensure that the software meets the determined criteria set forth in this document to that it may be pushed towards initial release.

Description of Test Environment

The software will be tested within the Android Studio SDK using the Android Emulator. The emulator uses an Android Virtual Device (AVD) which will allow us to configure size, Android version, hardware characteristics, etc. to simulate the software being run on different Android devices. Most if not all testing will occur on the AVD and we will conduct testing on physical Android devices if time permits. We (the developers) will be conducting the entirety of all testing performed on the software. At this time, we plan to conduct most if not all of the testing on Android Oreo 8.0 (SDK version 26) on a Google Pixel phone through the emulator. After completing the initial run of test cases, we will begin testing using other versions of Android (from SDK version 23 and above, as this is currently our minimum targeted SDK version). The software will operate on physical devices yet this in theory should not complicate testing if we are to only conduct testing through the Android Emulator.

Stopping Criteria

During testing, should a tester find an error, it will be documented and an attempt will be made to fix it. If the error cannot be fixed within a reasonable time frame (approx. 30 min. max), then the tester will revert any code changes made while attempting to fix the error and then resume with testing the section they are in charge of. After all testers have completed testing their sections, any remaining errors will be compiled together in a single document. After this, we will hold a group meeting to determine the current state of the software and whether we meet the criteria for delivery (outlined below) or must fix the errors and continue testing. If any errors remaining are *essential* (functional errors), then the software will not be considered deliverable and these errors will need to be fixed and re-tested. *Non-essential* (cosmetic) errors will not impact the software's deliverable status unless they severely impact the UX (User Experience).

In the event that no errors are found during testing, we will hold a group meeting and attempt to create a small set of new test cases to run. If no errors are found while running these test cases, then the software will be considered to be deliverable and testing will complete.

To determine if the software is deliverable, the following criteria must be met:

- More than 95% of the software is bug-free
- The software is compatible with the minimum supported SDK version up to the target SDK version
- There are no *essential* errors
- Any *non-essential* errors do not *severely* impact the UX
- All software components must correctly match the functionality defined in the Software Requirements Specification and by the client

---

Description of Individual Test Cases

Test Cases for Registration

No.	Test Objective	Test Description	Test Conditions	Expected Results
R.1	Verify text fields and buttons	Check placement of objects on screen	Android Emulator	Objects on signup page should be positioned correctly on screen
R.2	Check blank information	Leave text fields empty	Android Emulator	User should not be allowed to register and an error message should display
R.3	Check only required fields with acceptable data	Fill all required fields with acceptable data and click on signup button	Android Emulator	Account should be created in the system with all required information attached and user should be brought to the login screen
R.4	Check optional fields with required fields	Fill all required and optional fields with acceptable data and click on signup button	Android Emulator	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
R.5	Check password with insufficient strength	Fill password field with a string that does not contain numbers, letters, and symbols Password: 1a2b3c	Android Emulator	Proposed password should not be accepted by system and an error message should display
R.6	Check age field with non-acceptable age value	Input birthdate details that are under the age of 18	Android Emulator	Account should not be created and an error message should

		DOB: 1/1/2010		be displayed
R.7	Check invalid username	Input username that already exists in database into select username field Username: user123	Android Emulator	Account should not be created and an error message should display to screen

Test Cases for Login

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

Test Cases for Appointments

No.	Test Objective	Test Description	Test Conditions	Expected Results
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’ Name: “Test Service” Date: 12/15/2017 Time: 8:00 AM Price: $100	Android Emulator	Service should be created and stored on database linked to the specific doctor. It should be searchable from a patient’s perspective
A.2	Request an appointment	After following the steps described in M.1, click the first result and select ‘request appointment’	Android Emulator	The requested appointment should be added to the user’s ‘requested appointments’ page and to the associated doctor’s ‘view pending appointment requests’ page
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	Android Emulator	The details of the appointment (patient, doctor, service name, cost, date, and time) should be visible on a new page
A.4	Accept an appointment	Follow the steps detailed in A.3 and click ‘accept’	Android Emulator	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

A.5	Decline an appointment	Follow the steps detailed in A.3 and click ‘decline’	Android Emulator	A text field should appear where the doctor can provide a reason for declining the appointment request to the patient
A.6	Check conflicting appointments	While logged in as a patient with at least 1 appointment already scheduled, search for an appointment that occurs any time during the already scheduled appointment and click ‘request appointment’	Android Emulator	An error message should display indicating that the user is not allowed to create conflicting appointments. No change should be made in the database
A.7	Check appointment limit	Within the same 12 hour period, find 3 different services and request an appointment for all three	Android Emulator	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

Test Cases for Calendar

No.	**Test Objective**	**Test Description**	**Test Conditions**	**Expected Results**
C.1	Verify appropriate calendar for user	Login as any user and visit the calendar page	Android Emulator	The calendar with appointments tied to the specific user should be visible
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	Android Emulator	The appointment(s) for the user should for that specific date should display to the screen

Test Cases for Medical History

No.	**Test Objective**	**Test Description**	**Test Conditions**	**Expected Results**
MH.1	Upload medical history	Logged in as a patient at the "Medical History" portal, upload medical history Username: testPat Password: passwrd1!	Android Emulator	Created medical history should be uploaded to database and linked to the specific user's account
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: passwrd1!	Android Emulator	Appointment details should be added to the patient's appointment history
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'	Android Emulator	A PDF with the patient's medical history should download and become viewable
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	Android Emulator	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

Miscellaneous Test Cases

No.	Test Objective	Test Description	Test Conditions	Expected Results
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”	Android Emulator	A list pulled from the database should show all results related to “dentist”
M.2	Search for a service with empty string	While logged in as a patient, leave the search bar then search	Android Emulator	A pop-up should display stating that the search field was left blank
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	Android Emulator	Results that fit the filters should be the only ones to display
M.4	Check messaging feature	Follow the steps outlined in A.4, then visit the associated patient’s profile and select the ‘message’ button	Android Emulator	The messaging page should display and allow the doctor to send messages to their patient
M.5	Verify received message	Follow the steps outlined in M.4, then type the string ‘test’ and select send	Android Emulator	The message should only display in the patient’s ‘inbox’ as well as the doctor’s ‘sent messages’
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’	Android Emulator	The selected image should now display as the user’s profile image and be stored in the database

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	Firebase Project	The list of all usernames should be displayed along with their encrypted passwords
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	Google Pixel XL	A message should appear stating that the application is compatible with the device based on the manifest.xml document that accompanies the application
M.9	Verify device incompatibility	On a physical android phone running SDK 22, visit the application's page on the Google Play Store	Google Pixel XL	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

Trace of Individual Test Cases to Requirements

Test Cases	Corresponding Requirement(s)
R.1	3.1.1
R.2	
R.3	3.1.1
R.4	3.1.1 | 3.1.6 | 3.1.11
R.5	3.1.2

R.6	3.4.4
R.7	3.1.2
L.1	3.1.1
L.2	3.8.10
L.3	3.8.10
L.4	3.1.1 | 3.1.3 | 3.1.4 | 3.1.5 | 3.8.9 | 3.2.2.4 | 3.8.10
A.1	3.1.10
A.2	3.1.14 | 3.1.15 | 3.2.2.2
A.3	3.8.2 | 3.8.3
A.4	3.1.12 | 3.1.14 | 3.1.16 | 3.2.2.2
A.5	3.1.12 | 3.1.14 | 3.1.16 | 3.2.1.2
A.6	
A.7	3.4.5
C.1	3.1.8
C.2	3.1.9
MH.1	3.1.18
MH.2	3.1.19 | 3.2.1.3
MH.3	3.1.13 | 3.2.1.5
MH.4	3.2.1.5
M.1	3.1.17
M.2	
M.3	3.2.2.1
M.4	3.1.7 | 3.2.1.6
M.5	3.8.4 | 3.8.5
M.6	3.6.8
M.7	3.6.2

M.8	**3.3.1 | 3.3.2 | 3.4.3 | 3.9.1 | 3.9.2 | 3.9.5**
M.9	

Individual Requirement Statements

Below are the requirements included above and their individual statements. These requirements all originate from the SRS. Requirements not included below are either: too similar to already included requirements above, not measurable, or not necessary to be included for various reasons.

Requirement	**Statement**
3.1.1	The application login page shall allow user to create an account or to login to an existing account.
3.1.2	To create a new account, the user shall enter an available username and a password containing numbers, letters, and symbols when prompted.
3.1.3	The username and password shall be verified and checked against the database at login before the user is granted access to an account.
3.1.4	Users logged in as a Doctor shall be brought to the doctor-specific version pages in the app.
3.1.5	Users logged in as Patient shall be brought to the patient-specific version pages in the app.
3.1.6	The database shall encrypt user passwords before adding it to the system.
3.1.7	The system must allow a Doctor to send a message to a Patient and vice versa.
3.1.8	The system must allow the users to see their Calendar/Scheduling pages.
3.1.9	Clicking on a day on the calendar shall bring up a list of appointments for that day.
3.1.10	The application shall allow doctors to create services with time and cost as required inputs.
3.1.11	Each Doctor shall have specializations.
3.1.12	Doctors shall accept or reject appointments with a post-appointment note / rejection message.
3.1.13	Doctors shall have the option to download current or past patient history as a PDF.

3.1.14	Doctors shall receive a notification that a patient has requested an appointment and can choose to accept or deny.
3.1.15	Patients shall be able to request an appointment.
3.1.16	Patients will be notified that their appointments are either accepted or denied.
3.1.17	Patients shall have the ability to search for services by specialization, service type, and doctor.
3.1.18	Patients shall be able to enter in brief medical history (existing conditions, location, medication)
3.1.19	Once a doctor accepts an appointment, the service is added to the patient’s history.
3.2.1.2	A *decline* shall notify the patient and no new medical history shall be generated.
3.2.1.3	Upon *acceptance*, the patient shall be notified, and a new medical record shall be added to the patient's medical history.
3.2.1.5	If the patient has not blocked the visibility of her/his patient medical records, the Doctor, once acceptance has been made for an appointment, shall be able to view said patient's medical history.
3.2.1.6	Once an appointment has been accepted, the Doctor shall be able to communicate with the patient via a messenger application.
3.2.2.1	The patient shall be able to search for a Doctor based on, specialty, cost and appointment times.
3.2.2.2	The patient shall be able to request an appointment, the Doctor must first approve it before moving forward.
3.2.2.4	Each new user shall create an account (if they do not already have one)
3.3.1	The device shall have a screen size of at least 480 dp x 320 dp and must implement one of the supported logical Android framework densities (280, 320, 260, 400, 420, 480, 560, or 640 dpi).
3.3.2	The device shall have an absolute minimum of 424MB (if 280 dpi or similar) of memory available to the kernel and userspace (this value can increase depending on the specifications of the device).
3.4.3	The application will work within Android OS’s set up system, which allows users to set up their own default font and font size. This will help visually impaired users.
3.4.4	When registering new users, the application will make sure all users are 18 years of age or older to make sure children do not use the system to make unauthorized

	service agreements.
3.4.5	The system will make sure to limit how many appointments/services one user can make at a time, to prevent spam accounts misusing the system.
3.6.2	The system shall encrypt and decrypt passwords using Firebase authentication.
3.6.8	The system shall store images for profile pictures.
3.8.2	Appointment data in the database shall have a rule that only allows write access by the patient himself and his doctor for that specific appointment.
3.8.3	Appointment data in the database shall have a rule that only allows read access by the patient himself and any doctor he has had an appointment with previously UNLESS the patient has marked the private field of that record true then only the patient himself and the doctor he originally had his appointment with have read access.
3.8.4	Patient/doctor message data in the database shall have a rule that only allows read access by the patient/doctor that sent the message and the patient/doctor the message was sent to.
3.8.5	Patient/doctor message data in the database shall have rule that only allows write access once when the sender sends the message and allows no further modification.
3.8.9	User authentication shall be based on the combination of email address and password which is hashed with a cryptographic hashing function of MD5 strength or better.
3.8.10	User authentication shall be handled by the Firebase Authentication Authentication-as-a-service platform in combination with the Firebase Authentication Android library.
3.9.1	The application shall meet the requirements of the Android Minimum SDK Version stated in the manifest file.
3.9.2	The application shall meet the requirements of the Android Target SDK Version stated in the manifest file.
3.9.5	The application shall run successfully on all android devices running an Android Operating System version greater than the Minimum Target SDK Version stated in the manifest file.

Appendices