Sprint 3 Plan

Sprint Goal

The Sprint 3 goal is to implement more advanced functionality of the app that has been built previously, with calendar integration, sharing features, and notably notification features. This sprint is also fix and improve previous features according to stakeholder's feedback. The app is also created to be available to Apple App Store and Google Play Store

Sprint 3 Backlog (3 May - 24 May) V2.0

User Story ID	Tasks	Acceptance Criteria	Priority	Estimate Sizes	Dependencies
MS-2	• MW-147: Swap to encrypted protocol (HTTPS) DONE • MW-148: Create Enter Email to login endpoint DONE • MW-149: Add Forgot Password Functionality DONE • MW-170: Patient user registration page for first time user to fetch password DONE • MW-171: Patient register/forgot password page DONE • MW-181: Test refresh token fetching new JWT in mobile app and implement in all api calls DONE	1. Patients should be able to enter their email addresses to login for the first time or if they have forgotten their passwords 2. Users should be able to update their passwords 3. Users should not be logged out automatically for a full day 4. User data should all be encrypted whenever being sent on the network	Must Have	Large	
MS-3	• MW-151: Finalise Optional/Compulsory Genie Fields for File Uploads DONE • MW-153: Create Patient Account Status Endpoint DONE • MW-162: Create more robust error handling and prompts on html file upload page DONE • MW-182: Unit testing webapp - Patients page DONE • MW-183: Unit testing webapp - File upload page DONE • MW-152: Manage Appointment Deletions in Genie DONE	1. Admins should be able to upload a Genie file without all the correct fields and it not fail 2. Admins should be able to check a patient account's status 3. Users should see useful error messages when using the web apps 4. Patients should only see appointments that are still valid and in the future	Must Have	Large	

MS-4	MW-173: Implement admin account creation and deletion `DONE`	1. Admins should be able to navigate to an 'Admin Management' page 2. Admins should be able to view all admin accounts for the clinic 3. Admins should be able to create credentials for a new admin user 4. Admins should be able to delete the credentials of existing admin users	Must Have	Medium	
MS-10	• MW-43: Setup Backend Notification Service `DONE` • MW-45: Frontend Notification / UI `DONE` • MW-140: Automatically Send Appointment Reminders `DONE` • MW-141: Create Notification to Patient Endpoint `DONE` • MW-142: Send Messages to patients `DONE` • MW-143: Create Messages JPA Table `DONE` • MW-178: Handle notification permissions in mobile app `DONE`	1. Patients should receive notifications for upcoming appointments 2. Patients should receive notifications for messages from admins 3. Admins should be able to send patients messages and resources 4. Patients should be able to choose if they allow notifications or not	Should Have	Medium	
MS-11	MW-158: Add and test personal calendar integration for appointments `DONE`	1. Patients should be able to add their scheduled appointments to their native phone calendar app through an 'add to calendar' button associated with individual appointments 2. The appointment title and time/date should align with what is displayed in the MMS app and how it defaults in their native calendar app	Can Have	Medium	

MS-12	📗 MW-159: Create alternate appointment view - chronological list of appointments `DONE`	1. Patients should be able to toggle between calendar view and list view in the appointments page 2. In list view, appointments should be displayed in chronological order 3. Patients should be able to click on appointments to view their details	Should Have	Small	MS-3
MS-13	• 📗 MW-144: Create Generic Resources Template `DONE` • 📗 MW-164: Allow clinic to create and manage resources in webapp `DONE`	1. Admins should be able to send generic resources from templates instead of creating them from scratch	Must Have	Medium	
MS-14	• 📗 MW-157: Add personal patient notes feature for appointments `DONE` • 📗 MW-160: Display Genie appointment notes to patients `DONE` • 📗 MW-174: Change doctors and facilities address field on webapp to enforce a valid address `DONE` • 📗 MW-179: Remove doctor details from appointment details page - only need name and address `DONE` • 📗 MW-180: Improve default images on webapp - Doctors and Facilities page `DONE` • 📗 MW-184: Unit testing webapp - Doctors page `DONE` • 📗 MW-185: Unit testing webapp - Facilities page `DONE`	1. Patients should be able to make notes on an appointment 2. Admins should be warned if incorrect data such as invalid email is provided when creating new entities (doctors or facilities) 3. Users should have a positive experience with an easy to understand UI on the app and web app	Must Have	Medium	MS-3
MS-15	• 📗 MW-150: Create Email Service `DONE` • 📗 MW-161: Implement and test share button on appointments screens `DONE` • 📗 MW-163: Allow clinic to send patients messages in webapp `DONE` • 📗 MW-165: Allow clinic to send resources to specific patients `DONE` • 📗 MW-166: Allow patients to receive and view messages and resources sent to them `DONE`	1. Patients should receive an email at their address if they click forgot password or log in 2. Patients should be able to view all their messages in the app	Should Have	Medium	

| MS-1 | • 🗒 MW-167: Build app for Android app store DONE
• 🗒 MW-168: Build app for Apple app store DONE | 1. Patients should be able to download the app from the app store or google play store | Must Have | Medium | |

User Story ID	Priority	Estimate Story Points	Dependencies
🗒 MW-43: Setup Backend Notification Service DONE	Must Have	5	
🗒 MW-45: Frontend Notification / UI DONE	Must Have	2	
🗒 MW-140: Automatically Send Appointment Reminders DONE	Should Have	2	MW-43
🗒 MW-141: Create Notification to Patient Endpoint DONE	Should Have	2	
🗒 MW-142: Send Messages to patients DONE	Should Have	2	MW-43
🗒 MW-143: Create Messages JPA Table DONE	Should Have	1	
🗒 MW-144: Create Generic Resources Template DONE	Should Have	3	
🗒 MW-146: Write SQL Schema for automatic deployment DONE	Should Have	3	
🗒 MW-147: Swap to encrypted protocol (HTTPS) DONE	Should Have	3	
🗒 MW-148: Create Enter Email to login endpoint DONE	Should Have	2	MW-150
🗒 MW-149: Add Forgot Password Functionality DONE	Should Have	2	
🗒 MW-150: Create Email Service DONE	Should Have	4	
🗒 MW-151: Finalise Optional/Compulsory Genie Fields for File Uploads DONE	Should Have	2	
🗒 MW-152: Manage Appointment Deletions in Genie DONE	Should Have	3	
🗒 MW-153: Create Patient Account Status Endpoint DONE	Should Have	3	
🗒 MW-154: Backend Custom Error Handling DONE	Should Have	2	
🗒 MW-155: Backend Error Messaging (write copy) DONE	Should Have	2	
🗒 MW-156: Timezone handling/testing DONE	Should Have	2	
🗒 MW-157: Add personal patient notes feature for appointments DONE	Must Have	2	
🗒 MW-158: Add and test personal calendar integration for appointments DONE	Must Have	2	
🗒 MW-159: Create alternate appointment view - chronological list of appointments DONE	Should Have	1	
🗒 MW-160: Display Genie appointment notes to patients	Should Have	1	

DONE			
📄 MW-161: Implement and test share button on appointments screens DONE	Should Have	2	
📄 MW-162: Create more robust error handling and prompts on html file upload page DONE	Should Have	2	
📄 MW-163: Allow clinic to send patients messages in webapp DONE	Should Have	2	
📄 MW-164: Allow clinic to create and manage resources in webapp DONE	Should Have	2	
📄 MW-165: Allow clinic to send resources to specific patients DONE	Should Have	1	
📄 MW-166: Allow patients to receive and view messages and resources sent to them DONE	Should Have	2	
📄 MW-167: Build app for Android app store DONE	Should Have	3	
📄 MW-168: Build app for Apple app store DONE	Should Have	3	
📄 MW-169: Populate webapp home page DONE	Should Have	2	
📄 MW-170: Patient user registration page for first time user to fetch password DONE	Should Have	2	
📄 MW-171: Patient register/forgot password page DONE	Should Have	1	
📄 MW-172: Fix UI consistency between pages on the webapp DONE	Should Have	1	
📄 MW-173: Implement admin account creation and deletion DONE	Should Have	2	
📄 MW-174: Change doctors and facilities address field on webapp to enforce a valid address DONE	Should Have	2	
📄 MW-175: Patients can view doctor/ facility/ appointment locations on their native maps app DONE	Should Have	2	
📄 MW-176: Handle HTTPS requests for webapp DONE	Should Have	1	
📄 MW-177: Frontend time zone handling/ testing DONE	Should Have	1	
📄 MW-178: Handle notification permissions in mobile app DONE	Should Have	1	
📄 MW-179: Remove doctor details from appointment details page - only need name and address DONE	Should Have	0.5	
📄 MW-180: Improve default images on webapp - Doctors and Facilities page DONE	Should Have	0.5	
📄 MW-181: Test refresh token fetching new JWT in mobile app and implement in all api calls DONE	Should Have	1	
📄 MW-182: Unit testing webapp - Patients page DONE	Should Have	2	

🟩 MW-183: Unit testing webapp - File upload page `DONE`	Should Have	2	
🟩 MW-184: Unit testing webapp - Doctors page `DONE`	Should Have	2	
🟩 MW-185: Unit testing webapp - Facilities page `DONE`	Should Have	2	

Quick Link to Jira Backlog:

ME-Wombat

Jira