

Sprint 2 Plan

Sprint Goal

The Sprint 2 goal is to construct the base framework of the app, such as implement login authentication and user management, and the basic functionality of the app, providing appointment features and resources of information regarding the treatment and illness.

Sprint 2 Backlog (25 Mar - 2 May) V2.0

User Story ID	Tasks	Acceptance Criteria	Priority	Size Estimation	Dependencies
MS-2	<ul style="list-style-type: none">MW-29: Set up User Authentication DONEMW-31: Frontend/UI Login Implementation DONEMW-32: Backend Login Implementation DONEMW-34: Login Testing DONEMW-125: Create update password endpoint DONEMW-127: Remove Expired JWTs DONEMW-134: Add user ID to login return DONEMW-136: Frontend Mobile app Login auth setup DONE	<ol style="list-style-type: none">User should be able to find the "Log In" button easily on the app's welcome screen.The login screen should have fields for entering both username/email and password.Upon submission of valid credentials, the user should be granted access to the app's main interface.If invalid credentials are entered, appropriate error messages should be displayed prompting the user to re-enter correct details.The login process should be secure and protect user data from unauthorized access.Once logged in, the user should have access to all features and functionalities of the MMS app.The login process should be smooth and intuitive, providing a seamless user experience.	Must Have	Large	MS-3
MS-3	<ul style="list-style-type: none">MW-19: Add and remove patients DONEMW-36: Frontend/UI Patient Management Implementation DONE	<ol style="list-style-type: none">Mark should be able to log into the web-based backend of the Medical Secretary app without any issues.	Must Have	Large	Null

	<ul style="list-style-type: none"> • MW-37: Backend Patient Management Implementation DONE • MW-38: User Creation Functionality DONE • MW-39: User Deletion Functionality DONE • MW-41: Patient Endpoint Testing DONE • MW-95: Upload Patient File DONE • MW-100: Inherit Patient Users DONE • MW-108: Delete appointment status DONE • MW-111: Add test for appointment mapper DONE • MW-131: Frontend webapp patient page UI DONE • MW-133: Add get all patients endpoint DONE • MW-105: AppointmentController Test DONE • MW-106: FileController Test DONE • MW-114: Add test for user mapper DONE • MW-115: Add test for file service DONE • MW-116: Fix file upload to allow missing columns DONE • MW-129: Frontend webapp file upload page UI DONE 	<ol style="list-style-type: none"> 2. Mark should be able to navigate to the "Patients" section and perform actions such as adding and removing patients. 3. The system should add and remove patients when the Admin upload Genie HTML file with all necessary information. 4. Upon upload the new file, the system should update the patient list to reflect the removal immediately. 5. The system should allow HTML file with missing optional fields to still generate the necessary Patients and Appointments as long as the required values exist when Admin upload the Genie files. 			
MS-4	<ul style="list-style-type: none"> • MW-137: Frontend webapp login UI and auth DONE 	<ol style="list-style-type: none"> 1. Mark should be able to use the web app to log in as admin 2. The system should securely log admin securely 	Must	Medium	MS-2
MS-12	<ul style="list-style-type: none"> • MW-92: Frontend Appointment UI DONE • MW-101: Create SQL Appointment Table DONE • MW-107: Confirm or cancel appointment endpoint DONE • MW-119: Frontend Appointment-page Testing DONE 	<ol style="list-style-type: none"> 1. Ron should be able to access the list of appointments within the Medical Secretary app easily. 2. All past and future appointments should be displayed in chronological order, with past appointments appearing 	Should	Small	MS-3










		<p>before future appointments.</p> <p>3. The app should provide a seamless sorting mechanism to arrange appointments by date order.</p> <p>4. Appointments should be clearly labeled with their respective dates and times for easy reference.</p> <p>5. Ron should be able to navigate through the list of appointments effortlessly, ensuring a smooth user experience.</p> <p>6. The chronological order of appointments should remain consistent and accurate, reflecting any changes or updates made within the app.</p>			
MS-13	<ul style="list-style-type: none"> • MW-59: Backend Resource Management System DONE • MW-60: Frontend Resources UI DONE • MW-104: ResourceController Test DONE • MW-113: Add test for resource manager DONE 	<p>1. Ron should be able to access the resources and information links within the Medical Secretary app easily.</p> <p>2. The app should provide a diverse range of resources, including articles, videos, websites, and other educational materials.</p> <p>3. Resources should cover various medical conditions and treatment options relevant to Ron's healthcare needs.</p> <p>4. Links and resources should be updated regularly to ensure accuracy and relevance.</p> <p>5. Ron should be able to navigate through the resources effortlessly, with clear instructions provided within the app.</p> <p>6. The app should offer search and filter functionalities to help Ron find specific resources</p>	Must Have	Medium	MS-3

		related to his medical condition or treatment preferences.			
MS-14	<ul style="list-style-type: none"> • MW-96: Create Medical Institution DONE • MW-102: FacilityController Test DONE • MW-103: DoctorController Test DONE • MW-109: Get doctors for a patient DONE • MW-112: Add test for doctor mapper DONE • MW-120: Add get all facilities DONE • MW-128: Add get all doctors endpoint DONE • MW-130: Frontend webapp doctors page UI DONE • MW-132: Frontend webapp facilities page UI DONE 	<ol style="list-style-type: none"> 1. Ron can easily navigate through the app to reach the details. 2. Ron can find list of facilities and doctor that are available. 3. Ron can find doctor that is associated with appointment. 4. It should not be overwhelming in term of information. 	Must Have	Medium	Null

Tasks

Tasks ID	Size Estimation	MoSCoW Priority	Dependencies
MW-19: Add and remove patients DONE	6	Must Have	
MW-29: Set up User Authentication DONE	6	Must Have	MW-19
MW-31: Frontend/UI Login Implementation DONE	3	Must Have	
MW-32: Backend Login Implementation DONE	2	Must Have	
MW-34: Login Testing DONE	2	Can Have	MW-31, MW-32
MW-36: Frontend/UI Patient Management Implementation DONE	-	Must Have	MW-19
MW-37: Backend Patient Management Implementation DONE	0.5*	Must Have	MW-19
MW-38: User Creation Functionality DONE	2*	Must Have	
MW-39: User Deletion Functionality DONE	1*	Must Have	
MW-41: Patient Endpoint Testing DONE	1	Must Have	MW-36, MW-37
MW-91: Design Home page DONE	5	Must Have	
MW-95: Upload Patient File DONE	1*	Must Have	MW-19

MW-96: Create Medical Institution DONE	1*	Must Have	
MW-100: Inherit Patient Users DONE	0.5*	Must Have	
MW-101: Create SQL Appointment Table DONE	1*	Must Have	
MW-122: Frontend-mobileapp-Homepage-Tests DONE	4	Can Have	
MW-123: Setup WebApp Frontend DONE	2	Should Have	
MW-125: Create update password endpoint DONE	2	Must Have	
MW-127: Remove Expired JWTs DONE	0.5	Should Have	MW-29
MW-131: Frontend webapp patient page UI DONE	-	Must Have	
MW-133: Add get all patients endpoint DONE	0.5	Should Have	
MW-134: Add user ID to login return DONE	0.5	Should Have	
MW-136: Frontend Mobile app Login auth setup DONE	-	Should Have	
MW-137: Frontend webapp login UI and auth DONE	1	Must Have	
MW-59: Backend Resource Management System DONE	3	Must Have	
MW-60: Frontend Resources UI DONE	3	Must Have	
MW-92: Frontend Appointment UI DONE	-	Should Have	
MW-102: FacilityController Test DONE	1	Should Have	
MW-103: DoctorController Test DONE	-	Should Have	
MW-104: ResourceController Test DONE	1	Should Have	
MW-105: AppointmentController Test DONE	1	Should Have	
MW-106: FileController Test DONE	0.5	Should Have	
MW-107: Confirm or cancel appointment endpoint DONE	0.5	Should Have	
MW-108: Delete appointment status DONE	0.5	Should Have	
MW-109: Get doctors for a patient DONE	0.5	Should Have	
MW-111: Add test for appointment mapper DONE	1	Should Have	
MW-112: Add test for doctor mapper DONE	-	Should Have	
MW-113: Add test for resource mapper	1	Should Have	

DONE			
 MW-114: Add test for user mapper DONE	0.5	Should Have	
 MW-115: Add test for file service DONE	0.5	Should Have	
 MW-116: Fix file upload to allow missing columns DONE	1	Should Have	
 MW-119: Frontend Appointment-page Testing DONE	-	Should Have	
 MW-120: Add get all facilities DONE	1	Should Have	
 MW-128: Add get all doctors endpoint DONE	0.5	Should Have	
 MW-129: Frontend webapp file upload page UI DONE	1	Must Have	
 MW-130: Frontend webapp doctors page UI DONE	4	Must Have	
 MW-132: Frontend webapp facilities page UI DONE	1	Must Have	

Quick Link to Jira Backlog:



ME-Wombat



Jira