# Sprint 1 Planning Document

Han Quatro and the Chewbaccas

Darrin Landroche, Dara Lim, Nelli Hovhannisyan, Christian Barton

**Sprint Overview**

For our first sprint, we are focusing on getting the core framework assembled. This will allow us to add the actual functionality in coming sprints. By the end of the sprint we will have the views in place, but filled with fake data.

**Scrum Master:** Darrin Landroche

# Current Sprint Detail

**User Story #1**

Set up initial project

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task Description** | **Owner** | **Est. Time** | **Act. Time** | **Done (Y/N)** |
| **1** | Create Project | Christian | 30 min |  |  |
| **2** | Create Login App | Christian | 20 min |  |  |
| **3** | Create generic App for Doctor, Patient, and Secretary | Christian | 20 min |  |  |
| **4** | Push project to Git Repo | Christian | 1 hr |  |  |

Test case:

Verify files exist and that the server runs

**User Story #2**

Create generic views for patient

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task Description** | **Owner** | **Est. Time** | **Act. Time** | **Done (Y/N)** |
| **1** | View upcoming appointments | Nelli | 1 hr |  |  |
| **2** | Schedule Appointments | Nelli | 1 hr |  |  |
| **3** | View Records | Nelli | 1 hr |  |  |

Test Case:

Login as patient and navigate to each of the three pages.

**User Story #3**

Create generic views for secretary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task Description** | **Owner** | **Est. Time** | **Act. Time** | **Done (Y/N)** |
| **1** | Create Appointments | Dara | 1 hr |  |  |
| **2** | View Appointments | Dara | 1 hr |  |  |
| **3** | View/Edit patient info | Dara | 1 hr |  |  |
| **4** | Add/Remove patients | Dara | 1 hr |  |  |

Test Case:

Log in as Secretary and navigate between each of the four pages

**User Story #4**

Create generic views for doctor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task Description** | **Owner** | **Est. Time** | **Act. Time** | **Done (Y/N)** |
| **1** | Daily Calendar | Darrin | 1 hr |  |  |
| **2** | Add/Edit Patient Records | Darrin | 1 hr |  |  |

Test Case:

Log in as Doctor and navigate between each of the two pages

**User Story #5**

Design data models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task Description** | **Owner** | **Est. Time** | **Act. Time** | **Done (Y/N)** |
| **1** | User Models (Doctor, Patient, Secretary) | Christian | 3 hrs |  |  |
| **2** | Patient Information | Dara | 2 hrs |  |  |
| **3** | Schedule Information | Darrin | 2 hrs |  |  |
| **4** | Documents | Darrin | 1.5 hrs |  |  |

Test Case:

N/A, will be tested via implementation in future sprints.

**User Story #6**

Create CSS Standards

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task Description** | **Owner** | **Est. Time** | **Act. Time** | **Done (Y/N)** |
| **1** | Create Color Chart | Darrin | 1 hr |  |  |
| **2** | Sidebar component | Nelli | 30 min |  |  |
| **3** | Header component | Nelli | 30 min |  |  |
| **4** | Calendar component | Nelli | 30 min |  |  |
| **5** | Table component | Nelli | 30 min |  |  |

Test Case:

N/A, will be verified through the work done in future sprints.

**User Story #7**

As a user, I would like to be able to login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task Description** | **Owner** | **Est. Time** | **Act. Time** | **Done (Y/N)** |
| **1** | Create generic view for login page | Christian | 1 hr |  |  |
| **2** | Initially, just route based on login type given (ie. ‘secretary’, etc.) | Christian | 2 hrs |  |  |

Test Cases:

Verify that logging in to each of the three generic users works properly.

Verify that other inputs for the login route to failed login page.

# Remaining Backlog

**Non-functional requirements:**

-We will need a Dashboard/welcome page to route requests once logged in

-Encryption/Passwords (time permitting)

-Records and User information will be stored in a remote database.

-Implement Login view

-Implement Patient view

-Implement Secretary view

-Implement doctor view

-Implement database models

**Functional Requirements:**

-As a user I would like to be able to log in and log out

-As a doctor I would like to publish a record

-As a doctor I would like to edit published Records

-As a doctor I would like to view a patients published Records

-As a doctor I would like to add procedures that a patient can access

-As a doctor I would like to remove/edit available procedures

-As a secretary I would like to create appointments

-As a secretary I would like to edit/remove appointments

-As a secretary I would like to be able to view appointments

-As a secretary I would like to add patients (like admin)

-As a secretary I would like to archive/remove patients (Like admin)

-As a patient I would like to view my records

-As a doctor I would like to view my daily calendar

-As a doctor I would like to recommend a test or procedure

-As a patient I would like to request an appointment

-As a secretary I would like to update patient information

-As a secretary I would like to be able to confirm a requested appointment

-As a patient I would like to view my upcoming appointments

-As a patient I would like to remove any upcoming appointments

-As a patient, I would like access to multiple doctors

(i.e. Dermatologist, Dentist, etc) (time permitting)

-As a patient I would like to be able to receive appointment reminders

-As a secretary I would like to create billing statements/invoices