***Software Engineering***

***Software Requirements Specification***

***(SRS) Document***

**MedChart**

**September 16, 2022**

**Version 1.0**

**By: Bee Yang, Brandon George, David Amaya**

**I have abided by the UNCG Academic Integrity Policy**

|  |
| --- |
| **Table of Contents** |

TBD

[1.](#_heading=h.gjdgxs) Introduction 2

[2.](#_heading=h.30j0zll) General Description 2

[3.](#_heading=h.1fob9te) Functional Requirements 3

[4.](#_heading=h.3znysh7) Technical Requirements 3

[4.1](#_heading=h.2et92p0) Operating System & Compatibility 3

[4.2](#_heading=h.tyjcwt) Interface requirements 3

[4.2.1](#_heading=h.3dy6vkm) User Interfaces 3

[4.2.2](#_heading=h.1t3h5sf) Hardware Interfaces 3

[4.2.3](#_heading=h.4d34og8) Communications Interfaces 3

[4.2.4](#_heading=h.2s8eyo1) Software Interfaces 4

[5.](#_heading=h.17dp8vu) Non-Functional Requirements

6. Use-Case Model 5

1. **Introduction**
   1. **Purpose:** Allow employees to monitor patient room availability and manage patients information based on authorization of the employee and patients will be allowed to view their information (test results, diagnosis, etc.), contact medical staff, and schedule an appointment.
   2. **Document conventions:** The purpose of this Software Requirements Document (SRD) is to describe the client-view and developer-view requirements for MedChart. Client-oriented requirements describe the system from the client’s perspective. These requirements include a description of the different types of users served by the system. Developer-oriented requirements describe the system from a software developer’s perspective. These requirements include a detailed description of functional, data, performance, and other important requirements.
   3. **Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition. Acronym, Abbreviation** |
| CSS | A Styling language that is used to style web pages. We will be using this language to style MedChart |
| HTML | A programming language we will be using for the frontend of MedChart. |
| Java | A programming language created by Oracle. We will be using this language to build the majority of MedChart. |
| DB | An abbreviation for Database. |

* 1. **Intended audience:** This document is intended for the developers of the MedChart application (Brandon George, Bee Yang, David Amaya) and their supervisor (Sunny Ntini).
  2. **Project Scope:** The goals of the software are derived from the goals of the company in that the software directly follows the guidelines set by the business so that the software can provide the most benefit to the business while also assisting the consumers.
  3. **Technology Challenges:** TBD
  4. **References:** TBD

## General Description

* 1. **Product perspective:** Hospital management system:

Patients: Will include the patient's name, address, date of birth, phone number, emergency contact information, diagnosis, test results, and medication.

Medical Staff: Will be able to communicate with other medical staff through a message board, access patient’s information, manage the patient’s room number, and be able to contact the patient through message board.

Receptionist: Will be able to schedule appointments for patients, notify patients about upcoming appointments through message board, and manage appointments.

* 1. **Product features:** MedChart is a hospital management system application that will allow employees to monitor patients based on their room numbers. It will have a variety of information available to employees about their patients, such as diagnosis, medication, test results, etc. Additionally, the employees will have extensive information about the rooms that they are assigned to, such as vacancy, appointments, and type of room. Employees will also be able to utilize a time clock system to track their hours, as well as having a message board to notify them of patient updates. The motivation for MedChart is to simplify and streamline the hospital management system. MedChart will solve confusion in the hospital, allowing employees to manage hospital rooms and patients with ease.
  2. **User class and characteristics:** This application expects our users to have minimal knowledge and experience with computers. It focuses on the streamlining of hospital software so the care of patients can be more efficient. Most actors should become comfortable with our system in a short time, as it is as streamlined as possible.
  3. **Operating environment:** The software is designed to be operated in a hospital environment for better care of patients.
  4. **Constraints:** TBD
  5. **Assumptions and dependencies:** TBD

## Functional Requirements

* 1. **Primary**
* FR0: The system will allow the patient to view/edit their information, such as: address, date of birth, phone number, emergency contact information, diagnosis, test results, and medication
* FR1: The system will allow the patient to request medical attention from medical staff, such as:
* FR2: The system will allow the medical staff(nurses) to check the medical status of patients, such as: record vitals, give/edit medications.
* FR3: The system will allow the medical staff(doctors) to edit/update the medical status of patients, such as: give/edit diagnosis, prescribe medication.
* FR4: The system will allow the receptionist to check-in/assign a patient to a room.
* FR5: The system will allow the receptionist to contact patients, such as: notifying patients.
  1. **Secondary:** Some functions that are used to support the primary requirements.

## 4.Technical Requirements

* 1. **Operating System & Compatibility**
  2. **Interface requirements**
     1. **User Interfaces**

TBD

* + 1. **Hardware Interfaces**

The software is intended to run on and between multiple devices and multiple types of devices simultaneously. It can be run on mobile devices, tablets, or computers. The software does require a network connection, but it is not required to be fast.

* + 1. **Communications Interfaces**

The software supports communication between devices and individuals. Specifically between the patients, medical staff, and reception, all being able to receive and send messages when necessary.

* + 1. **Software Interfaces**

The software will be developed using HTML, CSS, and Java. The data will be stored in a local database at the hospital.

## 5. Non-Functional Requirements

* 1. **Performance requirements**

For Example:

The performance requirements need to be specified for every functional requirement. The rationale behind it also needs to be elaborated upon.

* NFR0(R): An expert patient will be able to login and view/update their information within 3 minutes.
* NFR1(R): An expert patient will be able to request medical attention within 1 minute.
* NFR2(R): An expert medical staff(nurse) will be able to check the medical status of patients within 5 minutes.
* NFR3(R): An expert medical staff(doctors) will be able to edit/update the medical status of patients within 5 minutes.
* NFR4(R): An expert receptionist will be able to check-in/assign a patient to a room within 3 minutes.
* NFR5(R): An expert receptionist will be able to contact patients within 3 minutes.
  1. **Safety requirements**

The software will conform to HIPPA regulations to protect all patient and employee data. Only verified users will have access to the restricted information.

* 1. **Security requirements**

Privacy and data protection regulations that need to be adhered to while designing of the product. For Example:

* NFR6(R): The system will authorize user to certain information based on their status in the system
* NFR7(R): Login ID, any user should have a login ID and Password to access the system.
* NFR8(R): Receptionists are allowed to view patient information, and add patients
* NFR9(R): Medical staff should be able to view and modify patient information and if the patient's information is modified it should sync.
  1. **Software quality attributes**
     1. Availability: The system will maintain availability at all times, as this software is extremely important at all hours and failure could result in lost lives.
     2. Correctness: The system will maintain correctness as much as possible, and will correct any issues that arise immediately.
     3. Maintainability: The system will be maintained at all times, as the software is critical for human lives.
     4. Reusability: The system will be designed for reusability in any hospital or medical center that may be able to utilize its properties.
     5. Portability: The system will be as portable as possible, allowing users to use it from almost any device or location.
     6. Flexibility: The system will be flexible, and can be corrected quickly if any issues arise.
  2. **Process Requirements**
     1. Development Process Used: The agile development process is being used to develop the software.
     2. Time Constraints: The prototype of the software will be completed by October 8th, 2022, and the final version will be completed on November 15, 2022.
     3. Cost and Delivery Date: The cost of the project will only be the time of the development team, and the delivery date will be November 15, 2022.
  3. **Other requirements**

These may include the legal requirements, resource utilizations, future updates etc.

* NFR10(R): The system will conform to HIPPA guidelines to maintain patient’s privacy.

## 6. Use-Case Model

Use Case Diagram:

Diagram

Description automatically generated

Use Case description

* Login/Logout - The user will be able to login to the system using credentials provided.
* Check-in - The receptionist will be able to check patients into MedChart, marking the patient presently waiting to be assigned a room.
* Contact Patients - The medical staff and receptionist will be able to send messages to patients through the provided messaging system.
* Assign Room - The receptionist will be able to assign available rooms to patients who have been checked in.
* View/edit Personal Information - The user will be able to access the personal information of the patients, which they can make changes to.
* Call for Assistance - The patient will be able to request assistance from medical staff.
* Check/Edit Medical Status - The medical staff will be able to view the medical status of the patient. From there, they can make edits/changes as necessary based on diagnosis, treatments, progress, etc..

Use Case Scenarios

Use Case Description - Scenarios

A) Login

I. Initial Assumption - The User has a registered account to login and patient information

saved in the database .

II. Normal - User enters their login information to login to their account

III. What can go wrong - Users forget their login information and have to request for their login information.

IV. Other Activities - User can request their login information and request to reset their is

login and can view their medical record, general personal information and request for

assistance.

V. System state on completion - User is login and can view their medical record, general

personal information and request for assistance.

B)Call for Assistance

I. Initial Assumption - The patient is logged in and would be able to request for assistance.

II. Normal - The patient makes a request for assistance.

III. What can go wrong - The staff not seeing the request from the patients.

IV. Other **Activities** - There are drop down boxes that show different types of requests the patient can make.

V. System state on completion - The request was sent to the staff and they saw the request, and helped the patient upon the request.

C) Assign Room

I. Initial Assumption - The patient is checked-in.

II. Normal - The receptionist assigns a room to the patient.

III. What can go wrong - The receptionist assigns an occupied room for the patients.

IV. Other Activities - The receptionist is able to see which room is occupied.

V. System state on completion - The patient is assigned to an empty room.

D)Check-in

I. Initial Assumption - The receptionist has a new patient waiting to be checked in.

II. Normal - Receptionist enters all relevant information for patient and saves it for future use.

III. What can go wrong - Receptionist enters incorrect information about patient.

IV. Other Activities - Receptionist can print and download information if needed.

V. System state on completion - Receptionist has completed check-in of patient and is returned to the main menu.

E)View/Edit Personal Information

I. Initial Assumption - User needs to view or edit personal information related to the patient.

II. Normal - User views or enters updated information about patient.

III. What can go wrong - User enters incorrect information about patient.

IV. Other Activities - User can print or download patient information.

V. System state on completion - User is returned to main menu and information has been viewed or updated.

F)Contact Patients

I. Initial Assumption - Medical staff or Receptionist is logged in and prepares to contact patient.

II. Normal - Medical staff or Receptionist enters message to send to patients and patient receives the message.

III. What can go wrong - Medical staff or Receptionist sends incorrect message or sends to incorrect patient; Patient doesn’t receive message.

IV. Other Activities - All users can view past sent messages to patients.

V. System state on completion - The Medical staff or Receptionist has sent the message and is returned to the main menu.

G)Check/Edit Medical Status

I. Initial Assumption - Medical staff is logged in and prepares to enter or view patient medical status.

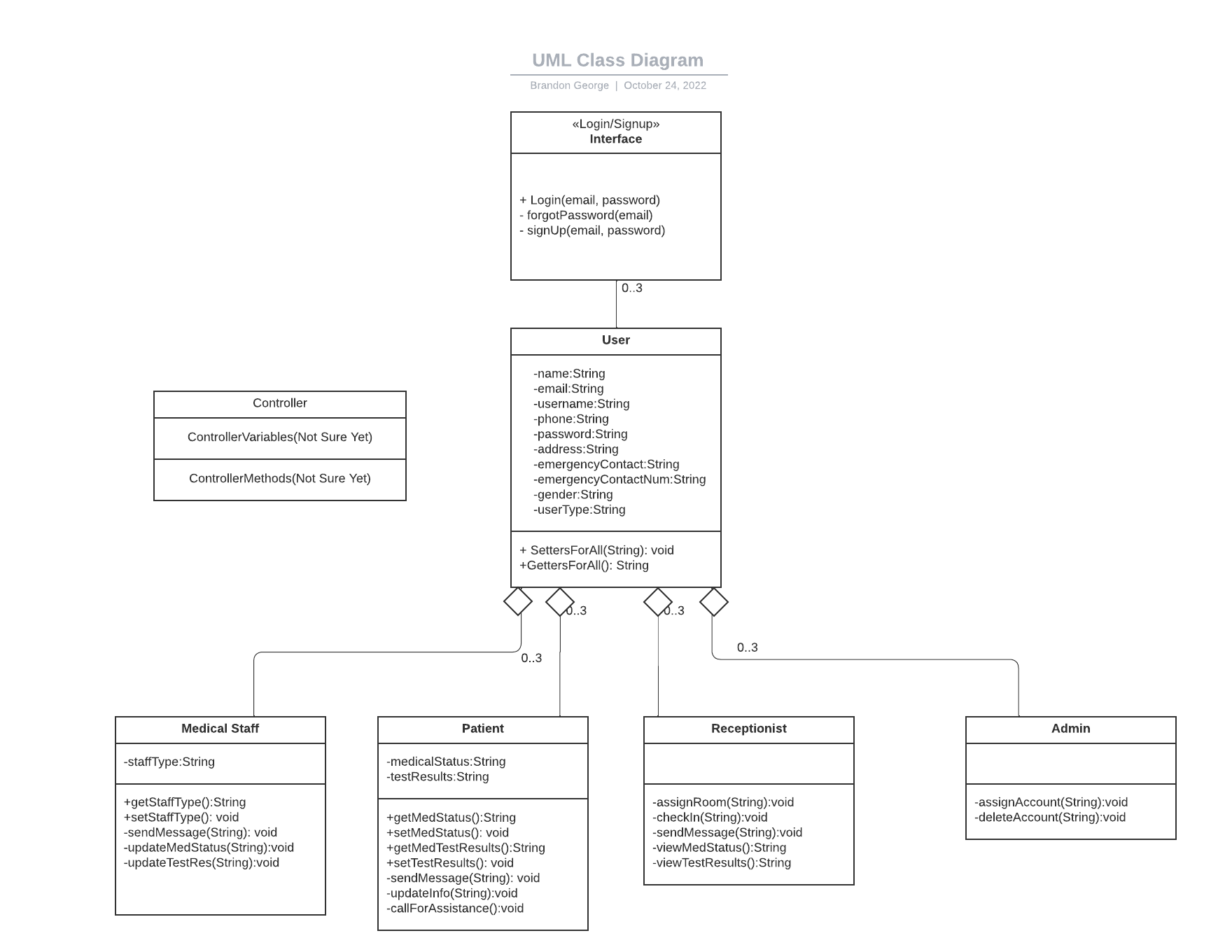
II. Normal - Medical staff enters or views status and saves or closes it out.

III. What can go wrong - Medical staff enters incorrect information.

IV. Other Activities - Medical staff can print or download info.

V. System state on completion- patient medical status is updated and staff is now back to the main menu.

## 7. UML Class Diagram



## 8. MVC Diagram

A picture containing qr code

Description automatically generated

## 9. State Machine Diagrams

Patient Diagram

Diagram

Description automatically generated

Medical Staff Diagram

Diagram

Description automatically generated

Receptionist Diagram

Diagram

Description automatically generated