Interface Requirements Document

**Table of Contents:**

[Preamble: 1](#_4bpno4ronun2)

[Technical Contents: 1](#_jzv34tsk6uwh)

[Version History: 2](#_f9xqu8xn00y6)

[Introduction: 2](#_z62n8s3a9wn7)

[Project Description: ​​​​​​ 2](#_ej9wd734vh5d)

[Use Case: 2](#_5xzqjcq8sm0r)

[UML/Process Diagram: 3](#_5i3t20qjgcc3)

[Key Features: 3](#_xglw1ywgvfvw)

[Client Needs: 3](#_2obsppt7u7uh)

[Competitive Analysis: 3](#_qlwuk3sc997x)

[Necessary Changes (V1->V2): 3](#_40cccvddland)

[Requirements: 3](#_u8u358byfy3a)

[Functional Requirements: 3](#_n8j1xivtblit)

[Non-Functional Requirements: 4](#_pp262i50wp5b)

[Glossary Index: 5](#_8i2rbc71hrda)

# **Preamble:**

## Technical Contents:

* Preface
  + Version History/Version Changes
* Introduction
  + Project Description
  + Use Case
  + Client Needs
* Glossary
* Requirements
  + Functional Requirements
  + Non-Functional Requirements
* System Architecture
* System Requirements Specification
  + Functional Requirements (detailed)
  + Non-Functional Requirements (detailed)
* System Models
  + UML Designs
  + Object Models
  + Data-flow Models
  + Semantic Models
* System Evolution
  + Fundamental Assumptions
  + Anticipated Changes
* Appendices
  + Hardware Descriptions
  + Database Descriptions
  + Hardware Requirements
  + Database Requirements
* Index
  + Alphabetical Index
  + Diagram Index
  + Function Index

## Version History:

* Vers. 1
  + This is the basic creation of the requirements document based on the gathered requirements from the customer(s). This document is seen as more or less as a note document to gather all requirements and ideas.
* Vers. 2
  + Reforming the requirements and notes into a more professional requirements document, i.e, this document.

# **Introduction:**

## Project Description: ​​​​​​

Our project is to recreate the interface for the Northern Psychiatric Services Website.

One of the key updates we will be making is the inclusion of a working calendar that will display and manage appointments. A big part of this will be the creation of a student and receptionist interface with different access to features. We also plan to create a note taking system for practitioners and students so they write down patient’s complaints.

## Use Case:

James, a practitioner for the NPS institution, ​​needs to be able to access their patients medical records, for an upcoming appointment. James goes onto the Web portal, and signs in with his credentials. From there, he can access any of his patients on a list presenting their basic information (*e.g.* name, age, sex *&c.*). Since James is a practitioner, he can access more detailed health records by clicking on a patient; and receptionists can access appointments and finances.

## UML/Process Diagram:

Text

# Key Features:

## Client Needs:

Northern Psychiatric Services needs to be able to, with their website:

* Set appointment date and time
* Choose preferred times of day
* Set type of appointment
* Have Separate patient records: Text
* Allow patients to be able to specify their preferred or previously visited doctors: Text
* Update patient information/records
* Send out a lab order
* Send out a prescription order
* Take notes while in an appointment

## Competitive Analysis:

* [Epic](https://www.epic.com/):
  + Easy to access and track patient’s charts.
  + Ability to use telehealth and e-visits(whatever they are).
  + Able to compare records with other organizations.
* Future plans to implement generative AI.
* Has mobile support
* Performance seems to be finicky.
* Can be difficult to change orders.
* Affiliated with many different organizations.
* Updates a lot.
* Can be customized.
* Difficult to learn.
* [Credible](https://www.qualifacts.com/credible-ehr/):
  + Atrocious design (in the client’s opinion)
  + Has mobile support
  + Easily configurable by the user
  + Includes “Faculty whiteboard and Duty Log” systems
  + Offers different variations, depending on business size (Qualifacts Insync, Qualifacts carelogic).
  + Does not hold previous notes for reference in patient/physician meetings.
  + Has had a reputation of going offline and not being able to go back online for upwards of half a day.
  + Gives access to most if not all of the information that the physician needs.

## Necessary Changes (V1->V2):

* Simplify & better organize UI elements/layout, over V1
* Replace client info from heading with logo/branding (No Logo, Northern Psychiatric Services).
* Add a homepage as a starting point
* Adjust client selection menu/list
* Implement a system to allow practitioners & medical students to take notes.

# Requirements:

## Functional Requirements:

1. Different users access different aspects of the website (Admin, practitioner, medical student, receptionist)
2. Practitioner requirements:
   1. View active/inactive problems
   2. View surgeries and past medical history
   3. Edit all patient records/info
   4. Take / view progress notes
   5. Send lab order
   6. Send prescription order (pharmacy integration)
   7. UI should be easy to use to allow for better communication between the practitioner and the patient
   8. Notes should be updated automatically to reduce repetition
3. Medical student requirements:
   1. Take / view progress notes
   2. View active/inactive problems
   3. View surgeries and past medical history
4. Receptionist requirements:
   1. View patient financial/insurance information
   2. Set patient appointments in 30 min increments (which patient with which practitioner)
   3. View basic patient info (only name, address, schedule, contact info, current medications, birthday, sex)
   4. Edit some info: (address, name, schedule, contact info, insurance info)
   5. Check-in patients when they arrive
   6. Notify the practitioner when one of their patients are there
   7. Add patient accounts if there are walkins
   8. Approve patient invoice upon check-out
5. Patient requirements
   1. View lab results
   2. View prescriptions
   3. View medical history
   4. View active/inactive problems
   5. View financial bills and make payments
   6. View upcoming appointments
   7. Book appointments/checkups
6. Admin requirements:
   1. All of the above
   2. Manage user accounts
   3. Change user passwords

## Non-Functional Requirements:

1. Info must be retrieved from the database in a timely manner.
2. Patient info is very private and can not be seen by outside sources

# Glossary Index:

| Word | Abbrev. | Meaning |
| --- | --- | --- |
| Version 1 | V1 | Refers to the original version of the website from the previous Software Engineering class. |
| Version 2 | V2 | Refers to the version of the website that we are currently working on. |
|  |  |  |
|  |  |  |