This is my amazing project document

# Introduction

The technology has become the necessary component in our daily life to accomplish various task. Facing this kind of pandemic that social distancing is essential the question is “How can front liner’s communicate with patients by doing distancing”, of course we can give a patient a form to fill-up every day then send a picture through social media. Still with the use of technology, what about if we create a system that could help the front liner’s and patients communicate, it is paper less at the same less hassle for the both parties. It could be an advantage to front liners because they can save their records through cloud storage, and for the patient because after filling-in the form they will just click submit rather than taking picture.

# Project context

We are currently facing the COVID-19 pandemic. Countries are struggling to identify hotspots for disease spread and implement quarantine measures while limiting the negative impact on society and economy.

Software can support some of the measures such as collecting daily reports of persons under monitoring to track common symptoms of COVID-19 and to identify individuals who should be quarantined.

# Purpose and description of the project

The purpose of the project is to implement a software system that lets local health professionals register persons who fall under monitoring regulations, issues identification codes for them, and keeps track of their submission of daily symptom reports until the end of the monitoring period.

As an additional function, the system could evaluate the likelihood of an actual infection with COVID-19 based on the reported symptoms. This has been implemented as a proof-of-concept without a claim to medical correctness.

The system is implemented in the C language and is driven by a simple console-based user interface based on the *curses* library. In a realistic setting, the system would most likely be implemented as a web application or mobile application. However, this was out of scope for the assignment.

# Objectives

* Provide a way for health professionals to register persons under monitoring.
* Issue a code for each person for identification purposes.
* Provide an entry form for persons under monitoring to submit their daily report.
* Give immediate feedback if quarantine is advised. (Experimental)
* Provide a comprehensive overview of a person’s reports to health professionals to assess the likelihood of an actual COVID-19 infection and the necessity of quarantine.

# Scope and limitation

## Scope

## Limitations

As described above, the system is currently implemented in C and it runs on a single computer. In a realistic setup, it would most likely be implemented as a web application to avoid the extra work of entering the submitted reports where the computer is located.

Also, the system does not implement any encryption of data. This would have to be added for a real system, as well as functions to change passwords and codes.

The system will not be usable for very large numbers of records, e.g. for a whole country, because it uses simple text files for storage instead of a database.

The immediate advisory for persons under monitoring is only added experimentally and would have to be verified by medical authorities. Also, the follow-up actions for persons under investigation are not yet implemented.

The system is implemented as a proof-of-concept and study example; some of the minor administrative functions have not been fully implemented, e.g. editing reports after submission, or deleting users and reports.

# Flowchart

# Output and user interface design

The system uses the console-based *curses* user interface library to present forms and menus so that users can easily navigate through its functions.

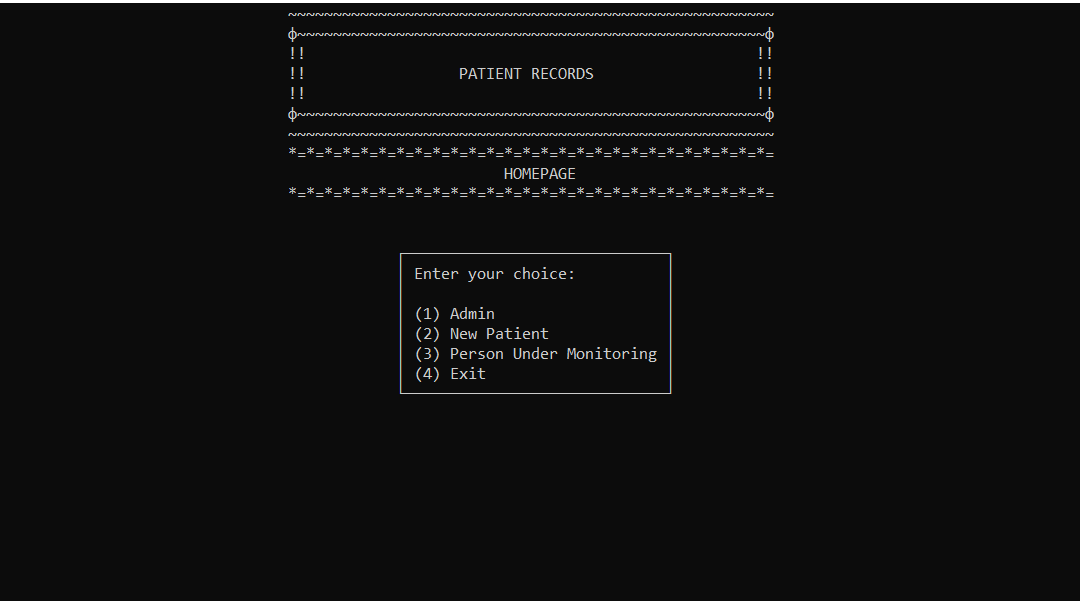
The following sections show screenshots of the major user interface elements, including a few examples of dialogs and messages that can appear during the use of the program.

The system does not produce any printed reports or outputs currently. All information is displayed in the user interface itself.

## Starting screen



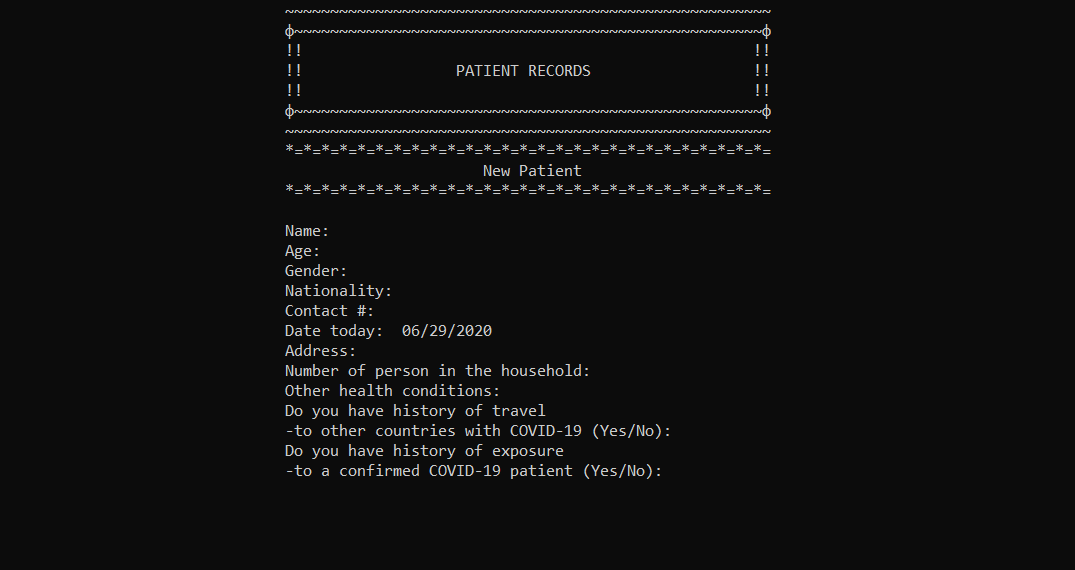
## Homepage



## Admin screen



## New Patient screen (User form)



## Person Under Monitoring screen (Report form)



## User information and selection screen



## Report overview screen



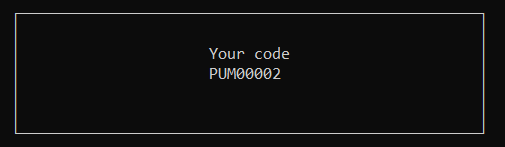
## Admin login pop-up dialog



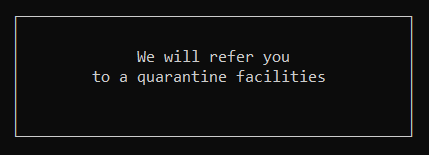
## User identification (patient login) pop-up dialog

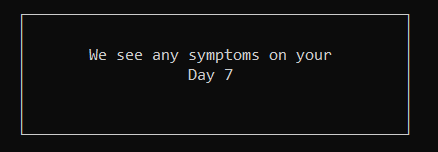


## Code issuing pop-up message



## Experimental advisory pop-up message





## Example of an informational message

Informational messages are meant to give the user guidance how to use the system.

