<u>Medecipher Solutions Project Proposal</u>
<u>Project Title: Nurse Schedule Builder User Interface</u>

Company description: Medecipher is a Denver-based health IT startup, funded by the National Science Foundation. Medecipher was started due to a lack of data-driven decision support tools for clinical operations leaders to use when designing, implementing, and adapting their care delivery models to ensure optimal outcomes for their patients, staff, and budget. We're passionate about giving operations leaders at health systems the tools they need to make data-driven and supported decisions about their capacity and resources – from predictive staffing levels to recalibration of bed allocation.

Background – staffing for COVID-19 times and beyond: Hospitals are under tremendous pressure to quickly respond and adjust their staffing to changing demand and capacity introduced by the COVID-19 pandemic. Adding to these challenges, the United States currently faces a worsening nurse staffing shortage expected to exceed half a million nurses by 2025, in large part due to retirement of the aging nursing workforce and influx of new patients into our health system. The burden of a growing resource shortage has already impacted nurses working in the profession as they mitigate the additional workload. The additional patient volume and rising shortage creates increased wait time for patients, higher risks of mistakes, and insufficient personal attention to patients in the hospitals. Prior to the pandemic, states such as California mandated patient ratios to protect their staff from burnout and turnover. Nurses were leaving their organizations in record numbers, with turnover rates as high as 40%, citing unsafe nurse staffing practices. Current strategies and practices for determining nurse staffing levels have proven inadequate to meet demand, and the problem continues to worsen. We are severely adding stress to the daily lives of our professional workforce.

While COVID-19 is the current crisis, the nurse staff shortage is not going away, and may only worsen and drive nurses into early retirement. Nurses who were already pushed to their max are now seeing higher than ever patient census and influx of new patients with little to no respite. They may consider retiring 5 to 6 years early. We expect permanent shifts in the way that hospitals recruit and retain their talent, and how they source their flexible staffing needs, and especially how they schedule staffers. The top priority for health systems is to protect and retain their workforce.

Innovation: Medecipher has created an ED nurse staffing recommender tool to match future nursing resources to future unknown patient service needs. The tool produces hourly ED patient census forecasts to compute hourly ED nursing demand. It adapts existing nursing schedules, using pre-approved modification constraints, to obtain hourly targeted nurse staffing levels that achieve the desired service-level based on management policies and practices. The system has three layers: (1) data science core that implements predictive modeling and decision recommendation technology engine; (2) middle layer designed and implemented for secure HIPAA-compliant data Input/Output; and (3) user interface.

Approach: Will adapt our existing tools ED nurse staffing tools to intake state and local data from surges such as COVID-19, and create further features and constraints to track burnout amongst ED nurses. **(1) Forecasting model**

Several hospital capacity models have been developed for COVID-19, including one for Illinois. We have not seen any for emergency departments. We will retrofit the models for ICU data to forecast patient demand. We will apply cluster analytics algorithms to find cohorts of similar patients with common staffing needs and create a common forecasting model for the cohorts. We will compute a common forecast for each patient cohort. Our approach will be a multivariate linear auto-regression with seasonality covariates that utilizes software packages such as "forecast", "rugarch" and "bft" available in the software system R. The forecast model will incorporate seasonality factors such as hour of the day, day of the week, and month of the year. We will apply ETS (Exponential Smoothing) and ARIMA (Autoregressive Integrated Moving Average). Moreover, other forecasting methods e.g. average, drift, and naïve will be applied based on the characteristic of the data. We will leverage variable imputation methods such as MICE (Multivariate Imputation by Chained Equations) to clean the data, impute missing values, and identify bad data through automatic use of threshold-based cut-offs.
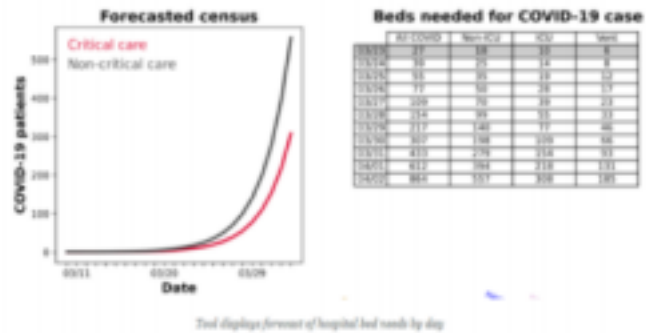

Figure 1: Displays the daily bed needs by bed type (non-ICU and ICU) along with the project number of vents for IL hospitals. Source

(2) Build a model to optimize and balance resource allocation across the ED

$$\min \quad \sum_{h=1}^{T}(p_h U_h + c_h O_h)$$
$$N_h + F_h + U_h + O_h = r_h,$$
$$h = 1, \ldots H.$$

Figure 2. Optimization model schematic for supply-demand balancing, where N_h: # RNs in the staffing grid, F_h: # RNs flexed, U_h: RN underage, O_h: RN overage, p_h: underage penalty c_h: overage cost, at hour h. Model is optimized across planning horizon H.

We will revise our preliminary optimization model to express supply-demand matching for ED nurses across all patient cohorts based on the staffing and service thresholds defined by ED operations leadership. The forecast from (1) becomes an input to the optimization model. The model incorporates shift types allowed by the hospital's position control (e.g. 8, 10, and 12-hour shifts) and allowed shift flexing actions (Figure 3). Values for Nh and Fh are generated from the forecasting model. The model minimizes understaffing and overstaffing at each hour, where under- and over-staffing is estimated based on a target ratio. Moreover, the

Shift:		Entire Shift		Start of Shift				End of Shift			
Start Hours:		Add	Cancel	-2 Hrs	-4 Hrs	+2 Hrs	+4 Hrs	-2 Hrs	-4 Hrs	+2 Hrs	+4 Hrs
0700	8	Y	Y	Y	n	Y	Y	Y	n	Y	Y
0700	12	Y	Y	Y	Y	Y	n	Y	Y	Y	n
0900	8	Y	Y	Y	n	Y	Y	Y	n	Y	Y
1100	12	Y	Y	Y	Y	Y	n	Y	Y	Y	n
1500	12	Y	Y	Y	Y	Y	n	n	n	Y	n
1700	10	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
1900	12	Y	Y	Y	Y	Y	n	Y	Y	Y	n

Figure 3. Acceptable flexing actions per shift type. e.g. for 0700 start 8-Hour shift, can reduce by 2 but not 4 hours.

model enables tradeoffs between the relative importance of underage (e.g. understaffing/safety risk) and overage (overstaffing/cost risk) at any hour. The model is calibrated to meet a safe staffing level desired by the management

(3) Interactive UI/Front-end
We will create functionality within the web-based application to assign users to shifts, track edits to the shifts, and track hours to flag nurses for potential burnout. Figure 4 is the current front-end, implemented using programming language AngularJS, that enables the ED nursing manager to view and adjust future daily scheduling recommendations under various demand scenarios.
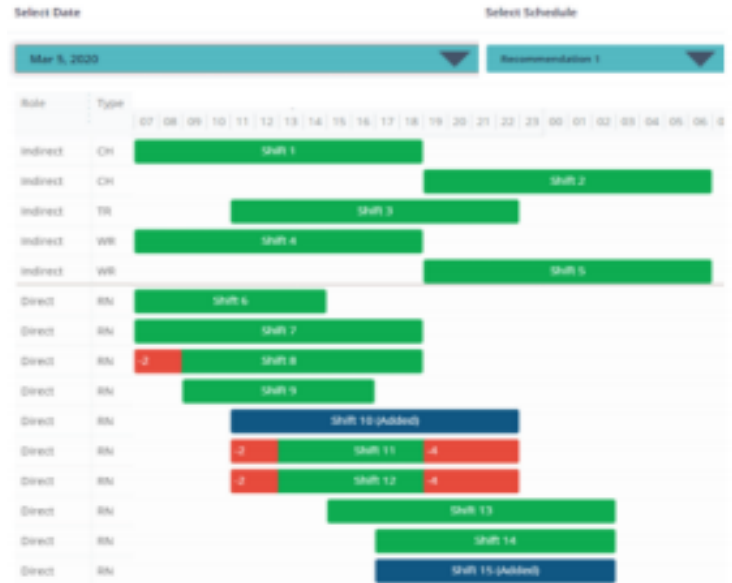

Figure 4. User Interface – optimal nurse schedule, shown for March 5, 2020.

Project Background: We have previously worked with a student team from Mines to create mockups for a new front-end UI. This UI is designed to allow our clients the ability to create customized schedules in reference to the output of our optimization models. The Medecipher team has finalized the mockup that we wish to implement for our website. Our goal is to implement this component in code to integrate it into our current website.

Proposed Solution: Transforming this static mockup into a finalized Angular web component.

Steps Involved:
1. Create an interactable prototype of the visualization through services such as *Marvel*, *Invision,* or *Figma.*
2. Once the prototype is approved, implement the prototype into a working *Angular* web component.
    a. Design Visualization
    b. Integrate Back-End/Data Flow
3. Integrate this component into the current Medecipher website.

Skills/Interests: This project will excite you if you have experience or an interest in
● Product Development

- ○ Mock up -> Final Implementation
- Web Design
  - ○ AngularJS
  - ○ Javascript / React
  - ○ HTML
- Back-End/Database
  - ○ Git/GitHub
  - ○ Creating Tables in PostgreSQL
  - ○ Establishing a connection between the Front-End and Back-End
  - ○ AWS CodeCommit

Desired Team size: 3-5
Front-End Development (2-3): Angular JS
Back-end Development (1-2): PostgreSQL, Container based services
3-5 team members, regular meetings with faculty, and frequent interaction with programmers to help overcome challenges.