Real-Time Demand Capacity Tool

Project Definition

Olivier Clermont (6445938)
Jonathan Ermel (6408238)
Mathieu Fortin-Boulay (6571541)
Philippe Legault (6376254)
Nicolas Ménard (6357275)

Presented to Prof. Liam Peyton
for the Software Engineering Capstone Project
SEG4910

University of Ottawa
January 23, 2015

Table of Contents

Project Title & Team

The application being developed will be referred to as **RTDC** (Real-Time Demand Capacity). Listed below are the team members collaborating on the project:

Name	Student Number	Email	Co-op Status
Olivier Clermont	6445938	ocler043@uottawa.ca	Non-co-op
Jonathan Ermel	6408238	jerme043@uottawa.ca	Non-co-op
Mathieu Fortin-Boulay	6571541	mfort059@uottawa.ca	Co-op
Philippe Legault	6376254	plega060@uottawa.ca	Co-op
Nicolas Ménard	6357275	nmena080@uottawa.ca	Co-op

Customer

The system will be developed for HealthNow. It will be used at the Montfort Hospital and the Queensway Carleton Hospital as part of a pilot study. Being in the healthcare business, it is their mission to provide excellent care and support to every patient. All this must be done in a timely fashion in order to encourage quick recoveries all while cutting down on operation costs.

Throughout the project, we will be interacting with Mr. Alain Mouttham (amouttham@rogers.com), chief technology officer of HealthNow. Mr. Mouttham holds a Master's in Computer Science and is co-founder of startups based in Silicon Valley and Ottawa. He also co-authored several healthcare studies and is now focusing on improving Patient Flow Management.

Objectives

The problem we aim to resolve, or help to resolve, is to ensure bed availability for all patients, every day, by the use of the RTDC method.

Briefly, the RTDC method is a method in four steps:

1. Assess the capacity (available beds + discharges by 2:00PM)
2. Estimate the demand by 2:00PM
3. Build and implement the plan to address the mismatch between the capacity and the demand
4. Evaluate the plan execution at the end of the day

In addition to the implementation of the RTDC method, we will implement unified communications (text messages and voice chat) within the app which will allow the nurses and the managers to communicate.

What motivate us to take this project is that it has the potential to improve the daily operations in hospitals by optimizing the patients' flows in the different units, and has the potential to benefit our society. Since HealthNow is a small smart-up enterprise, its CTO, Alain Mouttham, was seeking a team of students in order to develop a pilot study that will allow them to fully integrate the RTDC method in a bigger health related project.

Our main objective is to develop a system that will be used as a pilot study at the Queensway-Carleton Hospital and the Montfort Hospital, in order to evaluate the efficiency of the method. We will build the system, and give it to the client (HealthNow) when it will be completed so they can deploy it.

The project shall be deemed successful if it is in a state that the system can be deployed at a hospital, can be tested in real-life situations, and that the RTDC method is working as specified, meaning that it will help assess the capacity, estimate the demand, build and implement plans and evaluate plan executions for one or more patient flows. It is also expected that the unified communication with Asterisk will be working as expected.

System

Terminology

Listed below are the terms used to describe the system:

- **Action:** Task assigned to a nurse or a group that needs to be completed in order to discharge a patient. Each action is associated with a patient, the patient's room and needs for discharge.
- **Capacity:** Number of beds available.
- **Demand:** Number of newly admitted patients who require beds.
- **Discharge (DC):** Moment when a patient frees up a bed.
- **Group:** Collection of nurses or unit managers
- **Unit:** Set of rooms and their beds. Each unit has a capacity and a demand. A unit also has a list of actions requiring attention called Action Plan.

Use Cases

There are four main actors in the application, as illustrated in the use case diagram below. The first is the **nurse**. It is the actor responsible for completing actions so patients can be discharged.

The second actor is the **unit manager**. This actor is responsible for setting the demand/capacity for units and for creating an action plan for nurses to follow.

The third actor is a **stakeholder**, who can subscribe to and receive notifications related to actions that may concern him or her.

Finally, the last actor is an **administrator** who has the power to manage user accounts and units.

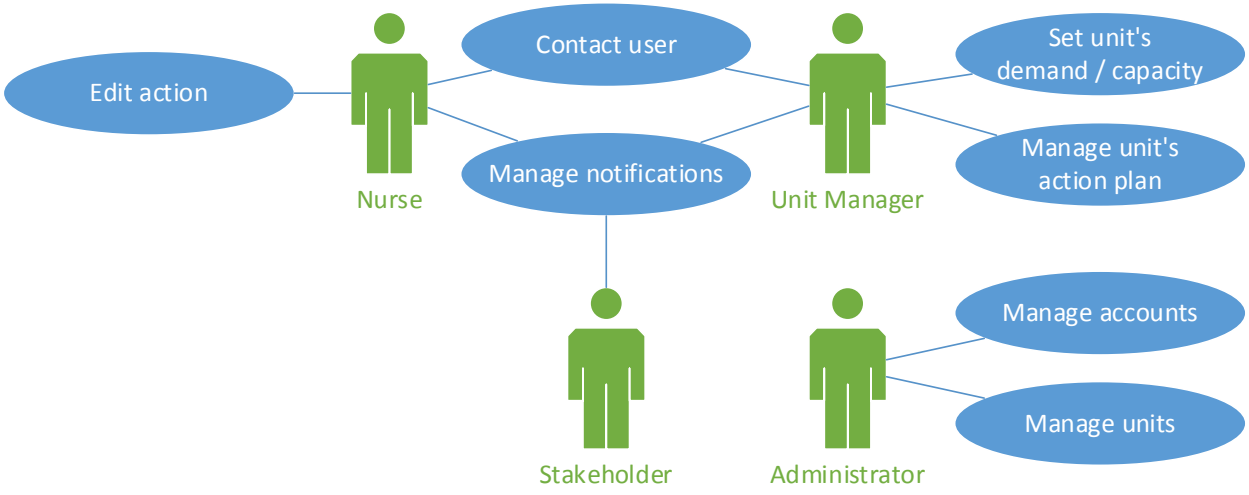


Figure 1 - Use case diagram

Edit action

A nurse can change the action that needs to be completed for a patient to be discharged. Alternatively, the nurse can mark an action as completed.

Manage notifications

A nurse or a unit manager can subscribe to and receive notifications concerning the completion of an action, new actions to be completed, new messages and missed calls. A stakeholder may subscribe to and receive notifications concerning the completion of an action and new actions to be completed. All the actors mentioned also have the ability to unsubscribe from said notification.

Contact user

Nurses and unit managers can contact one another through messages, voice calls or video calls.

Set unit's demand / capacity

A unit manager can set 5 values to help him determine a unit's capacity / demand. These values are the available beds, potential discharges, the discharges by 2:00 pm, the total admissions, and the admissions by 2:00 pm.

Manage unit's action plan

A unit manager can create a set of actions which need to be completed through the day for patients to be discharged. The action can be removed from the action plan once completed.

Manage accounts

The administrator can create, edit and delete user accounts.

Manage units

The administrator may create, edit and delete units.

User Interface Mockup

The following pages include a mockup of the user interface.

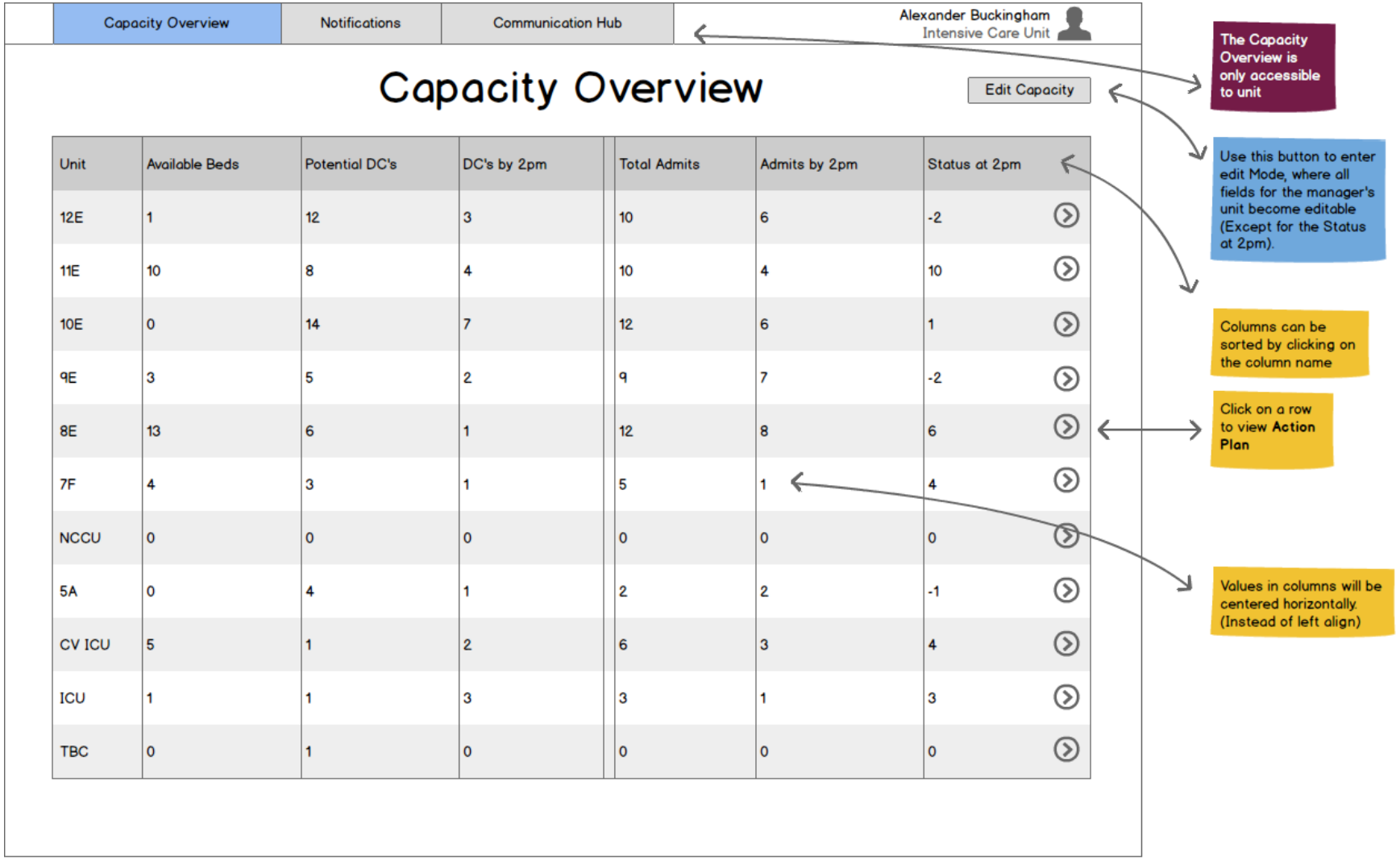


Figure 2 - Capacity Overview (Unit Manager View)

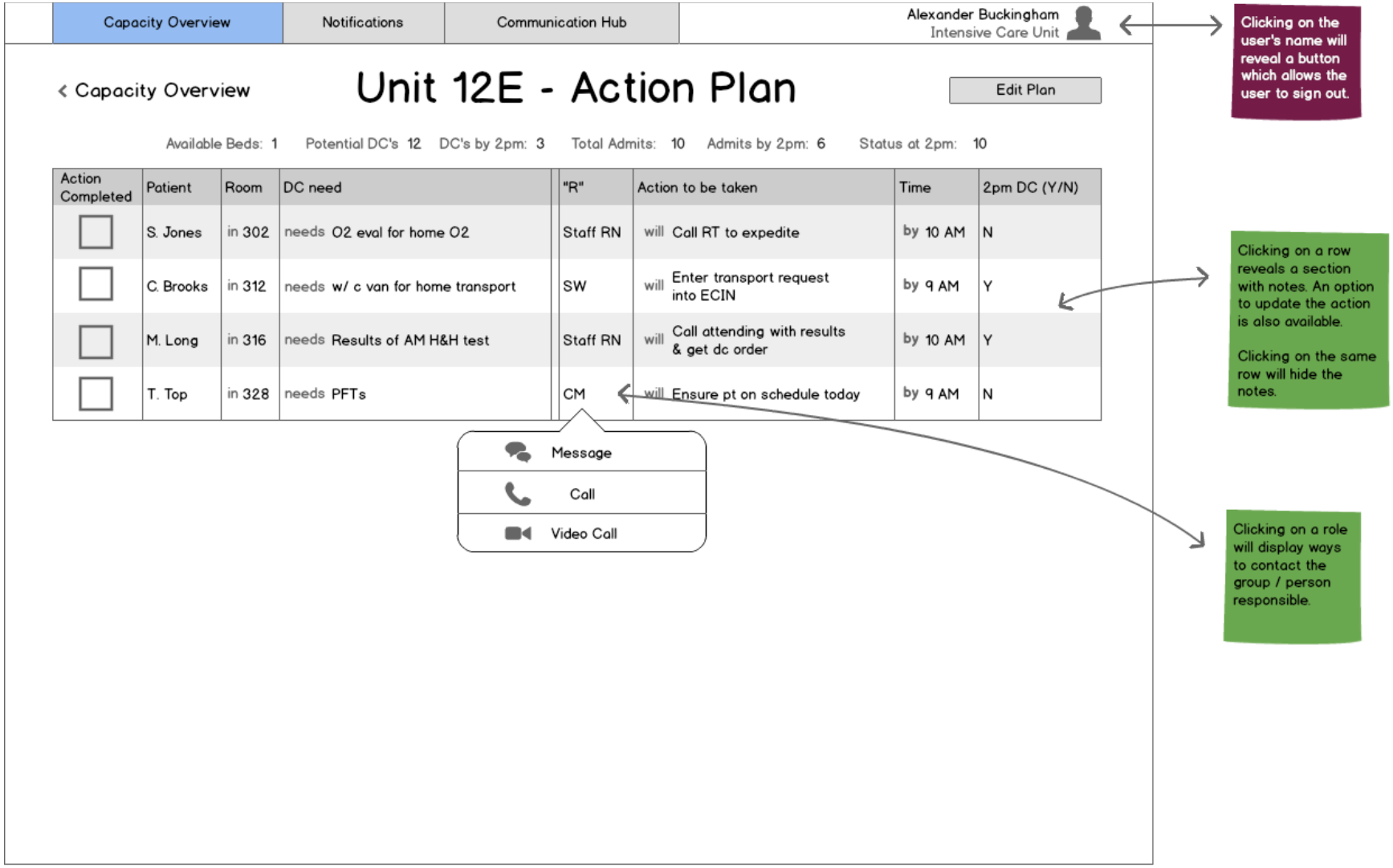


Figure 3 - Unit action plan (Unit Manager View)

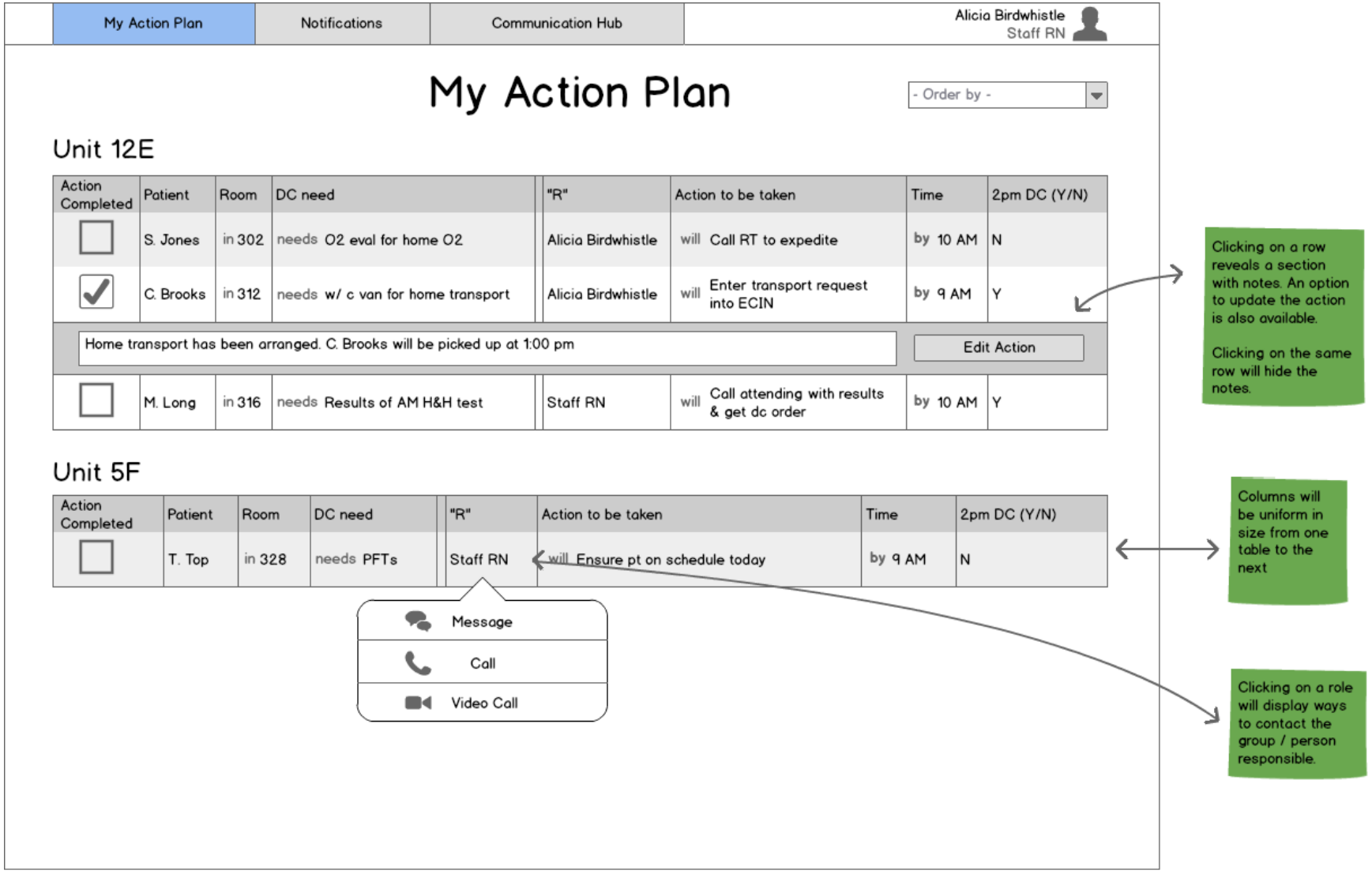


Figure 4 – Nurse's personal action plan (Nurse View)

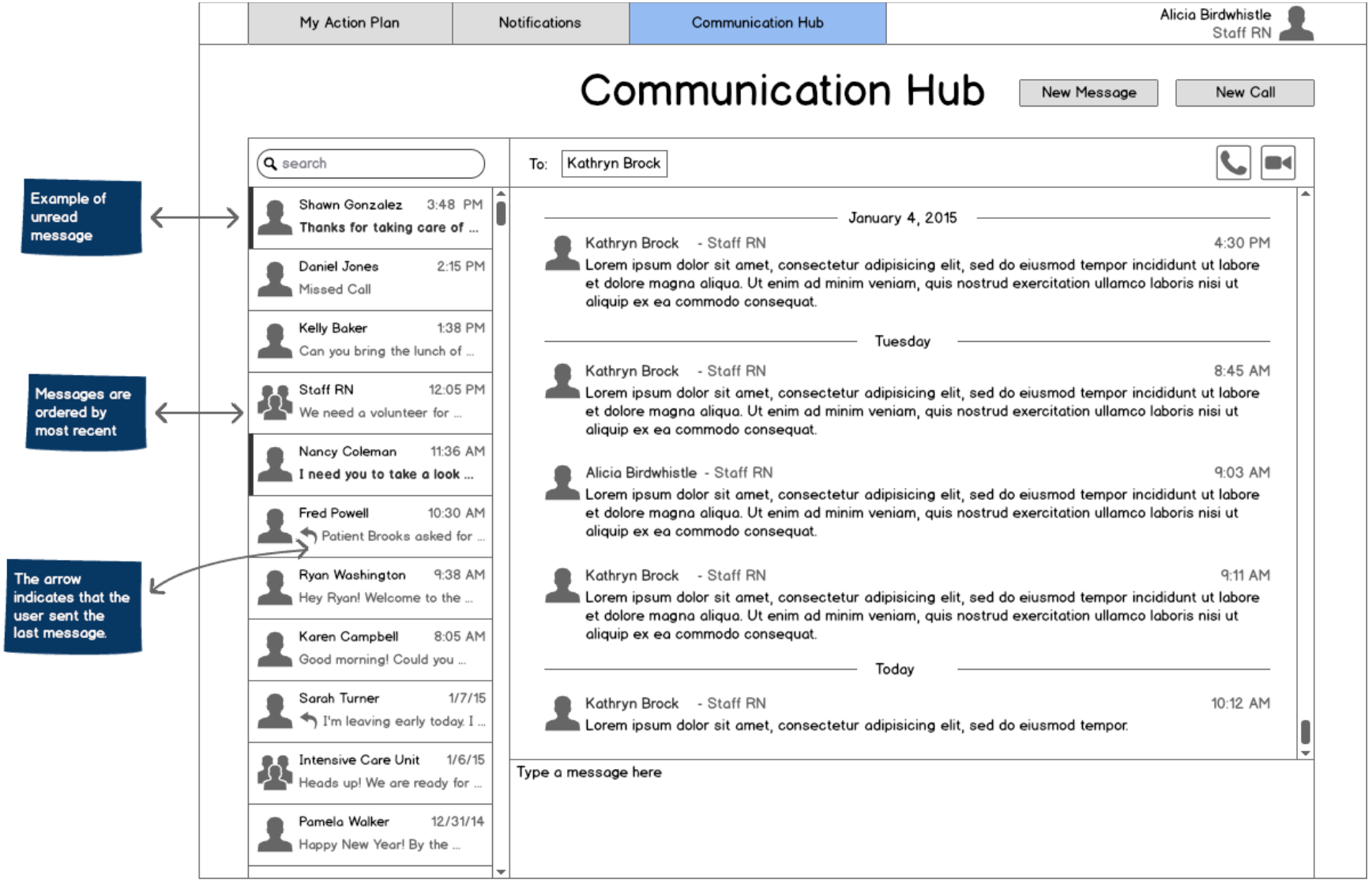
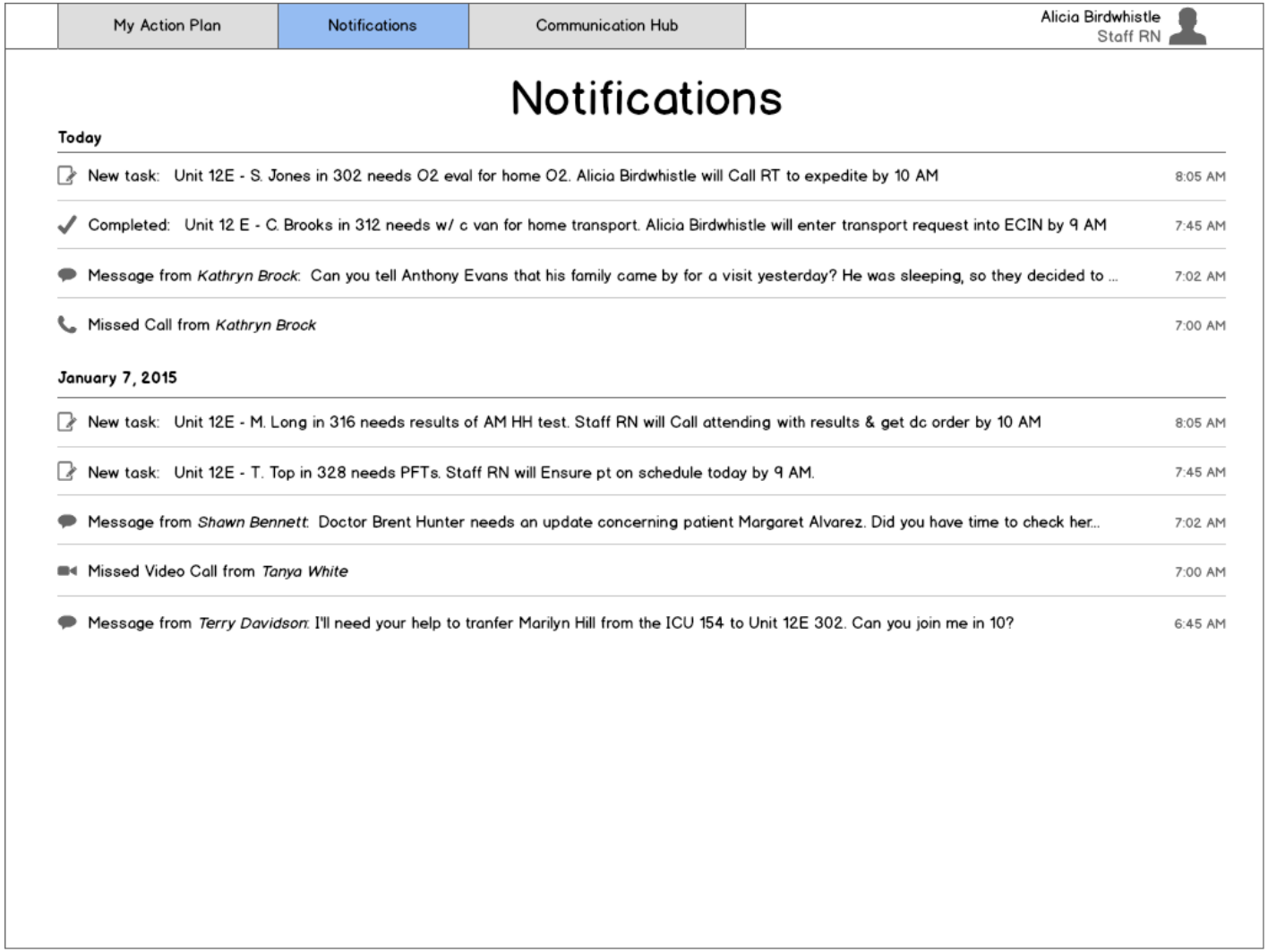


Figure 5 - Communication hub



Notifications are ordered with the most recent first.

Clicking on a notification starts an action:

- New Task / Completed - The user is brought to the task in question inside his / her action plan
- Message: The user is brought to the chat with the other user (inside the communication hub)
- Missed Call / Video Call: A new call is initiated to the other user

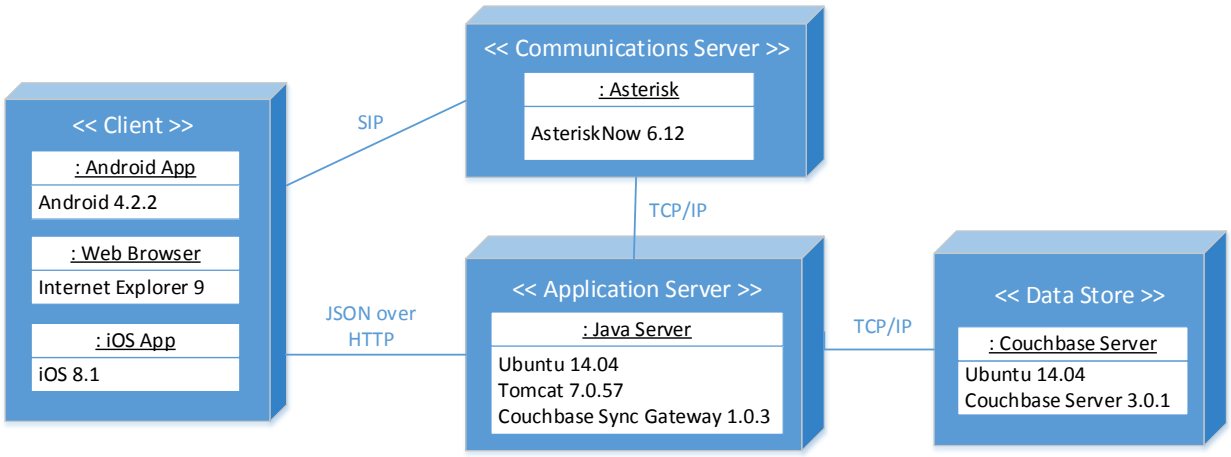
Figure 6 - Notifications

Architecture Description

The RTDC tool is a client-server system. As illustrated below, three different clients will be supported. However, the Android client will be the first one developed as it is the most commonly used by nurses. It will be tested on the Samsung Galaxy Tab 3 (8”), the tablet at our disposition. Afterwards, development will continue with iOS 8.1, which is second in popularity, and will be tested on the iPad mini 2 (7.9”). Finally, we will target Internet Explorer 9, because it is the only browser installed on the hospitals’ computers.

These clients will connect to two server components. The first is an AsteriskNow server which is at the core of the unified communications feature. AsteriskNow is a Linux distribution of FreePBX with the Asterisk software pre-installed. Asterisk is an implementation of a telephone private branch exchange (PBX). It uses the Session Initiation Protocol (SIP), and is mandated by the client. Asterisk is used in order to allow users to communicate within the app, by text messages or voice conversations.

The second is a Java server that hosts the remaining functions. This component interacts with the NoSQL data store known as Couchbase, which will store JSON documents. The Java server is installed on Linux, more precisely Ubuntu 14.04 LTS. Moreover, the machine will have Couchbase Server installed on it and it will act as our database. Couchbase provides us with few benefits. Firstly, its mobile component, Couchbase Mobile, will make the app more responsive and operable when there is a loss of connection. This is achieved by integrating Couchbase directly into our app. Couchbase Mobile acts as a local database, syncing periodically with the server. Secondly, Couchbase scales easily and uses NoSQL, which will allow us to have a more flexible data schema through our iterations. We also want to learn more about NoSQL databases.



Risk Assessment

Engineering Challenges

Multiple engineering challenges will need be solved throughout the development of the RTDC project:

- **Cross-Platform Support**: Developing native apps for Android and iOS, as well as a web interface, implies the use a variety of languages. It also extends the number of software versions to test against. In order to help manage this challenge, the team will focus on the development for specific versions / devices, as described below.
    - *Android 4.2.2 (Jelly Bean) – To be tested with the Samsung Galaxy Tab 3 (8")*
    - *iOS 8.1 – To be tested with the iPad mini 2 (7.9")*
    - *Internet Explorer 9*
- **Communication Framework Integration:** Taking advantage of Asterisk will require libraries which can communicate and exchange data with the framework. Being new to the Session Initiation Protocol (SIP) and the Asterisk framework itself, the team will have to quickly learn these new technologies.
- **Network Operations:** Couchbase will bring many benefits to the system such as dynamic schemas. However, conflicting changes will need to be resolved when synching data with the database. Moreover, the team will have extends its knowledge of databases to this new technology using NoSQL.
- **Ease of Use & Sturdiness:** The interface needs to be easy to use for users with no technical background. Otherwise, the application could cause hinder productivity and risks being rejected by the users. Furthermore, the application will need to be responsive, especially for network operations. Lastly, exceptions will have to be handled gracefully as the application should always be accessible.

Organizational Challenges

Undertaking a project of this size presents two major organizational challenges, as described below:

- **Schedule Management:** The customer and team members all have different availability. Thus, meetings will have to be well planned to take efficient use of everyone's time.
- **Centralization of Information:** Information pertinent to the project is spread out to a variety of sources such as websites, emails, text messages, etc. Presentations, documents, code and all other information should be accessible in one place.

Deployment Challenges

Demonstrating the progress of the application to the customer presents a few challenges:

- **Hosting:** Throughout the project, database and server components will be installed a machine connected to a private home network. Thus, for demonstration purposes, the machine will have to be configured in such a way that it is accessible from the university and other external networks. This problem should not apply to the final product as the application is meant to be used within a private network.
- **Mobile App Installation:** While Android offers a variety of options for distributing apps for free, iOS requires a paid license. For private distribution (outside the App Store), an Enterprise

developer license is necessary. However, project files can be packaged and shared for demonstration using a Mac and the iOS Simulator (free of charge).

Impact Assessment

Legal Issues

The use of the application falls under Ontario's Personal Health Information Protection Act of 2004 (PHIPA). In short, the collection of personal information requires consent and shall be limited to the strict minimum necessary. Furthermore, security safeguards need to be put in place to protect the information.

While the RTDC tool facilitates the execution of tasks towards a patient's discharge, a discharge order is still required as per regulation 965, section 16 of Ontario's Public Hospitals Act of 1990 (PHA).

The RTDC project source code will be the intellectual property of the student team listed in the *Project Title & Team* section of this document. The students retain the right to reuse any code as they please, and continue the development of the application once collaboration with the customer is over. Nevertheless, the customer will have access to the code to continue developing the system for production use. The customer may share the code with partner Openface, or the hospitals involved in the pilot study for that purpose.

Liability Issues

Offences to the Personal Health Information Protection Act can result in fines of up to $50,000 for a convicted individual, or $250,000 for corporations. Similarly, offences to the Public Hospitals Act can result in fines between $50 and $1,000 for a convicted individual.

The software is provided "as is". In no event shall the authors, the RTDC team, be liable for issues arising from the use of the software.

Societal Issues

The RTDC project has the potential to be very beneficial to society as it may help hospitals increase bed availability. Consequently, hospitals could see reduced wait times and patients will have a lower risk of health complications while waiting to be discharged.

User Community

The intended user community, the hospital staff, may see an increase in productivity and efficiency in their work.

Financial Impact

The design and development of the RTDC project will be done free of charge before the end of 2015. Afterwards, support, maintenance and updates through the RTDC team may be done at a fee (to be negotiated). Resources (e.g. servers, tablets, iOS Developer Program membership) for the development of the application will be provided by the RTDC team and the University of Ottawa's Engineering Faculty.

The estimated costs of running and maintaining the system is $10,200 per month. This includes the salary of 2 developers and server costs.

An hour in a hospital's medical unit costs around $20. In contrast, waiting for over an hour in the emergency room costs $1,000, seeing as the risks of infection increases as time goes by. So, if a single bed could be freed up 4 hours earlier than usual, the hospital could save up to $4,080 in a day. If the application allows 5 beds to be freed up 4 hours earlier each, hospitals have the potential of saving $20,400 in only a day. Assuming the total cost of developing the application is $400,000, the hospital could see a return on investment in less than a month.

Project Plan

The project roles have been distributed as follow:

- **Project Manager:** Nicolas Ménard
- **Business Analyst:** Olivier Clermont, Philippe Legault
- **Test Manager:** Jonathan Ermel
- **Build Manager:** Jonathan Ermel
- **Architect:** Mathieu Fortin-Boulay
- **Lead Developer for Android Component:** Olivier Clermont
- **Lead Developer for iOS Component:** Philippe Legault
- **Lead Developer for Web Component:** Olivier Clermont
- **Lead Developer for Application Server:** Mathieu Fortin-Boulay
- **Lead Developer for Asterisk:** Nicolas Ménard
- **Lead Developer for Database:** Mathieu Fortin-Boulay

The team will follow the Scrum development framework with 4-weeks sprints. In other words, working software will be delivered at the end of each sprint along with the corresponding documents required for each iteration. The primary focus of the first 3 iterations will be creating a fully functional Android client. Afterwards, support for iOS and the web will be added to the system. The project manager will oversee the progress of the project and write the various project status reports. Here is a detailed overview of the progress planned for each sprint (iteration).

Sprint 1 – Android Demand/Capacity, Accounts & Project Definition

The development of the Android client has begun. Administrators can create accounts and units. Unit managers can set their unit's demand/capacity.

The project manager is responsible for the project definition document.

Sprint 2 - Android Action Plan, Notifications & Requirements Analysis

Unit managers can manage action plans. Nurses can change actions or mark them as completed. All users can subscribe to and receive notifications.

The business analyst is responsible for the analysis report.

Sprint 3 – Android Communications & Quality Assurance Plan

Users can communicate amongst each other, thus making the Android client complete.

The build manager is responsible for the quality assurance presentation.

Sprint 4 – iOS Demand/Capacity, Action Plan & Architecture Design Report

The development of the iOS client has begun. Users can manage action plans, mark actions as completed, change actions and set a unit's demand / capacity. User accounts can be created as well.

The architect is responsible for the design report.

Sprint 5 – iOS Notifications, Communications & Quality Assurance Report

The remaining functionality is added to the iOS client. In other words, users can communicate amongst each other from the iOS client. They can also subscribe to and receive notifications.

The test manager is responsible for the quality assurance presentation.

Sprint 6 – Web Client & Final Report

The web client has been fully implemented, meaning the project is complete. The project has been packaged and handed to the customer.

The project manager is responsible for the final report.