**EXECUTIVE SUMMARY**

The Electronic Pre-Operative Anesthetic Plan (EPAP) is a proposed native mobile application prototype for the Android and iOS platforms. This application is designed to increase efficiencies between anesthesia healthcare professionals, technicians and pharmacists in an operating room (OR) setting.

The EPAP application enables anesthesia professionals to create and communicate an anesthetic plan consisting of limited relevant patient data, operation procedural details, medication, and equipment. This plan will enable credentialed anesthesiologists to review the plan, designated pharmacists to fill requested medications earlier, and designated technicians to prepare the operating room with the necessary materials and equipment.

The system will produce benefits including reductions of procedure delays due to unavailable medications or materials, reduction of medication and material waste, and increased efficiency in the provisioning of high-demand equipment.

The core team progressed through the first three phases of the project, including initiation, planning, and design during the first semester. With the guidance of the project sponsors, the team created a thorough list of requirements for the mobile application, concisely summarized above. Subsequently, the project charter collected these requirements and outlined the vision, project scope, assumptions, risks, and management plans. The team defined the project plan, use cases, and the proposed technology stack for development during the planning phase. Prototypes and a thorough data model with relevant information and connections were made during the design phase.

The team utilized the time between semesters to review a self-designed curriculum intended to acquaint the members with the technologies and programming languages to be used in the coding phase, including JavaScript and React Native.

Implementation began at the beginning of the second semester, which involved setting up application development through GitHub and Atom. Development itself was organized into eight phases, which each focused on different core functionality of the app. Back-end development was handled through Kumulos, a service which contained the database and facilitated the relevant API calls. The final application as developed by the team includes functionality to create new and manage existing users, login, add/edit/delete equipment, drugs, and plans themselves, and message other users with comments and concerns. Simultaneous testing was also conducted during the duration of development and documentation was also created. Full handover is expected to occur at the end of the semester, in May 2018.

**PROJECT OVERVIEW**

The Electronic Pre-Operative Anesthetic Plan (EPAP) is a native mobile application prototype for the Android and iOS platforms. This application is designed to increase efficiencies between anesthesia healthcare professionals, technicians and pharmacists in an operating room (OR) setting.

The EPAP application enables anesthesia professionals to create and communicate an anesthetic plan consisting of limited relevant patient data, operation procedural details, medication, and equipment. This plan enables credentialed anesthesiologists to review the plan, designated pharmacists to fill requested medications earlier, and designated technicians to prepare the operating room with the necessary materials and equipment.

The system creates benefits including reductions of procedure delays due to unavailable medications or materials, reduction of medication and material waste, and increased efficiency in the provisioning of high-demand equipment.

The team largely communicated through Basecamp, a project management tool containing relevant documents, schedules and meeting appointments, checklists, and chat and collaboration functionality.

Several use cases for the main users of the application, including administrator, physician, pharmacist, and technician, were generated for the application. Additionally, a detailed data model indicating the data to be processed by the application and their relationships were also created. A functional prototype was created using a service called Mockingbot and included the basic screens which were expected to be in the final application. Kumulos, a mobile backend as a service (MBaaS) utilized by the project, had two-fold purpose of serving as a database and handling API calls.

The application itself was developed using React Native for both iOS and Android platforms. The final application as developed by the team includes functionality to create new and manage existing users, login, add/edit/delete equipment, drugs, and plans themselves, and message other users with comments and concerns. Related data is stored and modified in the Kumulos databases on the back end of the application.

**PROJECT STAKEHOLDERS**

Project Team Members

* Ashley Reese, Project Manager & Lead Developer
* Ben Pitts, Front End Developer & Test Analyst
* Daniel Crittenden, Assistant Project Manager & Back-end Developer
* John Peeler, Business Analyst & Full-Stack Developer
* Justin Ucol, Full-Stack Developer & Database Administrator

Sponsors

* Dr. Aalap Shah, MD, Subject Matter Expert, Cedars-Sinai Hospital
* Dr. Vikas O’Reilly-Shah, MD, PhD, Technical Liaison & Subject Matter Expert, Emory University

Project Advisors

* Dr. Mark Huber, PhD, Terry College of Business, University of Georgia
* Dr. Nikhil Srinivasan, PhD, Terry College of Business, University of Georgia

**PROJECT SCOPE STATEMENT**

The following deliverables are considered to be within the scope of the project:

* The project team will create a prototype Electronic Pre-Operative Anesthetic Plan (EPAP) native mobile application that is able to run on iOS and Android platforms and that meets the requirements specified in the Project Charter.
* The application will pass user testing by the project stakeholders identified in this document.
* Where reasonable, the prototype application will be designed with consideration for:
  + Health Insurance Portability and Accountability Act of 1996 (HIPAA) compliance
  + Fast Healthcare Interoperability Resources (FHIR) compatibility
* The team will create user manuals, application documentation, and thorough code comments to be delivered to the project sponsor upon conclusion of the project.
* Basic user training will be provided to project sponsors.
* A transition plan will be created for the purposes of turning over the application and underlying codebase to the project sponsors upon conclusion of the project.

Unless specified, all other items are considered to be outside of the project scope, including:

* The project deployment into a production environment.
* The ongoing maintenance, development, and updates to the application.
* Full integration of the application with electronic health records (EHR).
* Assurance that the application is HIPAA-compliant or that the application meets any other medical-related government regulations.
  + Note: per the above “in-scope” statement, the team will make all reasonable efforts to build a strong foundation for future HIPAA compliance. However, the project team is unable to provide assurances that the prototype will meet HIPAA or other medical-related regulations.
* A web-based version of the application.
* Functionality that provides calculations for weight-based dosing.

**IMPLEMENTATION**

*Strategy/Approach*

Initially, the team intended on implementing the development of the application into a single phase with coding, testing, and deployment occurring subsequently. The implementation plan was changed early on to a more Agile approach using phased team sprints. Development was divided by functionality into their own phases. Within a particular functionality phase, the tasks were further subdivided into specific assignments. Testing also occurred simultaneously and direct feedback from this type of testing helped to fine tune the application along the way. The team eventually settled on a modified version of the phased team sprints, however. The team itself was subdivided into development groups who would work on each phase together with the advantage of multiple developers checking work as it was done.

The planned phases were per the following list:

1. Login & User Profiles
2. Admin Features (Create and register users)
3. Admin Features (Add supplies and drugs)
4. Physician Features (Plan CRUD)
5. Pharmacist and Technician Interactions
6. Advanced Interactions (System auto archive and physician clawback plan)
7. Day of Surgery (DOS) View
8. Request System (Transfer plans and plan suggestions)

Overall, the change in approach helped to reduce development time and progress could be more easily tracked with the assignments through Basecamp. Testing throughout rather than after development also helped to ensure the application code remained reliable with the addition of new functionality and features as development progressed.

*Challenges*

A significant challenge the team encountered was related to working with new coding languages and frameworks. The team has had some instruction on coding and web development during the course of the MBT program, and some members have even more experience than that level. However, React Native was a tool unfamiliar to all of the members. There was some base familiarity due to the JavaScript involved, but there were many more differences when applied to React Native.

Usage of mobile backend as a service (MBaaS) was another challenge. Aside from being unfamiliar with MBaaS as a concept, there was the additional learning curve of the selected service. Kinvey was the service chosen by the team initially, but its lack of support for React Native was a significant issue. Kumulos proved to be a decently powerful alternative with React Native support and it provided the necessary backend features for the application.

Minimizing merge conflicts when combining together coded work was another challenge encountered by the group. With multiple developers and branches of work off the same code base, it was difficult at times to control merge conflicts. In order to address this issue, one team member was designated to handle all of the merges with another team member serving as an assistant. This approach proved to be directionally more successful.

The last major challenge was related to scheduling meetings with the sponsors. It was a challenge in itself to find times which worked for both the core team and sponsors due to very busy existing schedules. While several group meetings were successfully scheduled, other solutions to communicate and discuss ideas and problems were employed. Smaller group meetings versus full group meetings were favored to help ease scheduling constraints. Videos as a tool to demonstrate new functionality was also used, which had the advantage of demonstrating on a “watch when you can” basis.

*Solutions*

*Lessons Learned*

Communication was one of the biggest learnings as a result of this project. Initially, the team discussed with the sponsors ideas for functionality, which involved the Mockingbot prototype as well as sponsor sketches. However, there seemed to be a difference in opinion in what exactly a prototype was and what it included. Having a better understanding of the sponsors’ immediate goals and definitions, including asking clarifying questions and making better use of additional wireframes and use cases, could have better shaped our work and approach to the solution.

Considering the development performed and the framework used, a better and more realistic estimate of expected time per phase and task could have been achieved. The team felt it underestimated the learning curve of the technology used in this project. Becoming more familiar with these technologies and allowing for more time to familiarize with the tools would have helped to make development smoother and easier.

One other lesson learned involves minimizing the scope of the project. It may have been more beneficial to better define the specifics of the deliverables. In addition, a better identification of what the minimum viable product entailed would have helped moderate the scope of the project. Finally, starting on development even earlier than allotted for this project would have helped the team deliver a more complete form of the application.

*Required Follow-Up Actions*

There are some features which did not fall under minimum viable product (MVP) but are still incomplete as of the final presented version. These items include:

* Edit plan functionality is incomplete
* Notifications and messages tab does not have error handling in place in the event a plan is removed or discarded
* Users cannot view themselves in the user listing
* Users cannot deactivate or revoke administrative privileges for themselves

The project sponsors have been made aware of these defects and outstanding features during the project handover.

**CONCLUSIONS AND FINAL THOUGHTS**

Over the course of two semesters, the project team created a functional application which meets MVP.