Project Proposal for the
Google Summer of Code Program

Operation Theater Module

from Lukas Johannes Breitwieser

March 21, 2014

Contact details	Address	10 Ortweinplatz, Graz, Austria
	Phone	+43-664-386-20-64
	E-Mail	lukas.breitwieser@gmail.com
	Skype	lukasbreitwieser
	IRC	lbat

Mentor: Harsha Kumara

OpenMRS

Contents

1 Project proposal

The operating theater (OT) represents one of the most critical hospital units, both in patient safety and financial terms. The correct management of operating theaters represents an important step towards meeting the urgent need to deliver high quality care with limited resources. [Agnoletti et al, 2013, P2] This document proposes a module to complement the functionality of the EMR-System OpenMRS to achieve these goals.

1.1 General aspects

Figure 1 shows the life cycle of a surgical record. It is divided into three stages (preoperative, perioperative and postoperative). After each phase the user triggers a function to proceed to the next one. The business logic verifies if all requirements are met and notifies the user if something is missing (e.g. Patient consent for the surgery is missing.) The individual steps of the workflow inside the OT are shown in figure 2.

Managing the processes inside a surgical department is an immensely difficult task. On the following pages I did my best to share my vision of how an operation theater module for OpenMRS could look like. Of course, I know that I have only addressed a few issues.

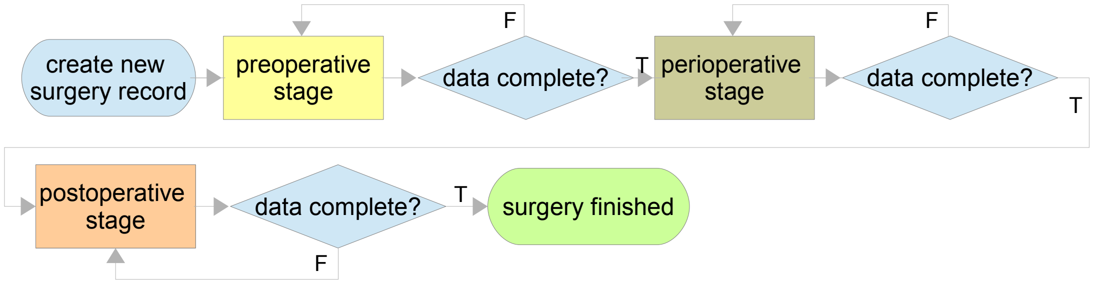


Figure 1: Life cycle of a surgical record

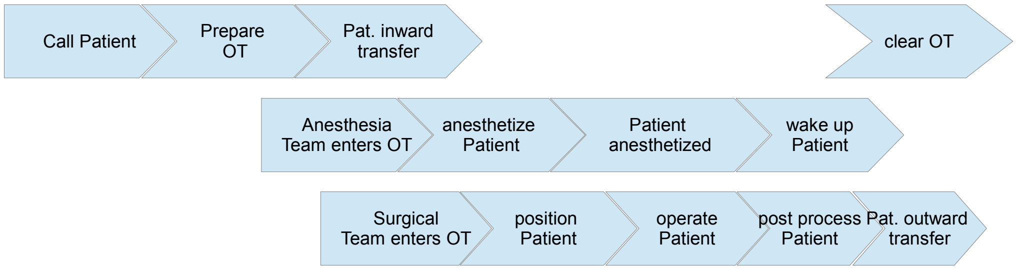


Figure 2: Subprocesses inside the operation theater – based on [Welk and Bauer, 2006, P.40]

1.2 User Interface

The layout in figure 3 shows an overview of the most important parts of the OT module. The home screen consists of a minimalist design with access to the most important actions, as in the new OpenMRS-Design:

- Find Surgery Record
- View Scheduling
- Create new Surgery Record
- Administration

To avoid a crowded confusing start screen, administrative actions are moved to a dedicated page. Master Data opens a page where the user enters data about operation theaters, procedures and resources. Inside the Report view, the user can execute various analysis and track the performance. The warehouse is needed to cover the demand of consumable supplies for the scheduled surgeries.

1.2.1 Surgery Record View

On the top of the page, general information about the ID of the patient and the planned surgery is displayed. The most important data from a medical viewpoint is constantly shown on the left. This includes allergies, long-term medication, pre-existing illnesses and other information the physician wants to have in his field of view. The remaining space is filled with a tabbed pane. It splits the data that has to be managed into three categories. The patient pane displays all information that will be gathered during pre- and postoperative stages. The remaining perioperative view gives access to all actions and information which will be needed inside the operation theater. Administrative entries that are also important for scheduling can be found under the Tab "General". Also the content of the discharge letter, that will be written at the end of the entire process, is viewed there.

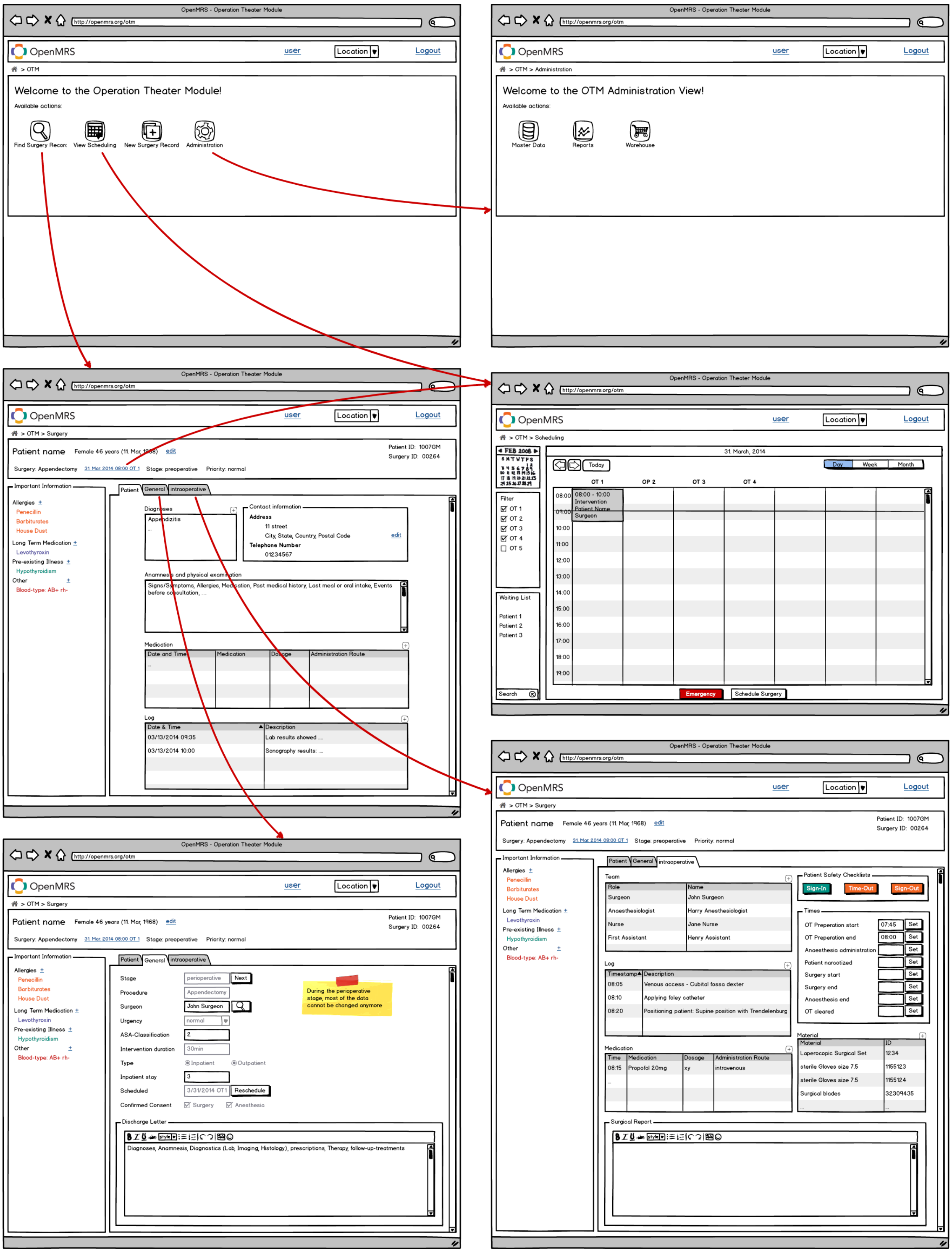


Figure 3: User Interface – see Appendix for larger images

1.2.2 Scheduling View

The scheduling page shows a summery of the planned surgeries, within the selected time frame. At least a daily view should be available. UI-Elements to quickly switch between dates are important. Two major actions include the normal scheduling of a surgery from the waiting list as well as the scheduling of an emergency (see section 1.3.2). A valuable feature would be to filter the displayed entries according to the operation theater, surgeon or procedure.

1.3 Scheduling

Scheduling is an important part to reach a high degree of resource utilization. It is also a very complex process that involves a lot of data and constraints.

1.3.1 Automatic scheduling

The following questions have to be answered during the process:

- How much time will be required to do the surgery?
- How much time will be required to prepare and clean up the OT?
- How urgent is the surgery?
- How long is the patient already waiting for the surgery?
- Has the surgery already been postponed?
- Is a bed required – inpatient vs. outpatient?
- How long will the bed be required?
- Is an intensive care bed required?
- Find a free OT
- Is there a free (intensive care) bed for the required time?
- Does the surgeon have another surgery scheduled at the same time?
- Are all required materials and equipment available?

If all the questions above can be answered, an effective automatic scheduling will be possible. There are some time estimations included in the above list. This means that at the end of the process the actual times should be compared to the initial estimations to improve quality.

1.3.2 Emergency management

Another very important task is to alter the plan if unforeseen emergencies occur. At first, one has to answer the question, whether there should be unscheduled times during the day that could compensate for the extra time needed. This will of course depend on the surgical department.

Four different kinds of emergencies will be distinguished:

- Emergency - surgical intervention without delay! - e.g. rupture of inner organs
- Urgent - the next finishing OT will be used - e.g. open fracture
- Deferred Urgent - search for a free OT on the same day e.g. abscess
- Elective - surgeries that are planned more than 24h in advance

[Welk and Bauer, 2006, P. 159]

Actions will be very different, depending on this classification. The less severe the emergency, the more flexible is the scheduling. The flow chart in figure 4 shows the steps for finding an OT for an Emergency.

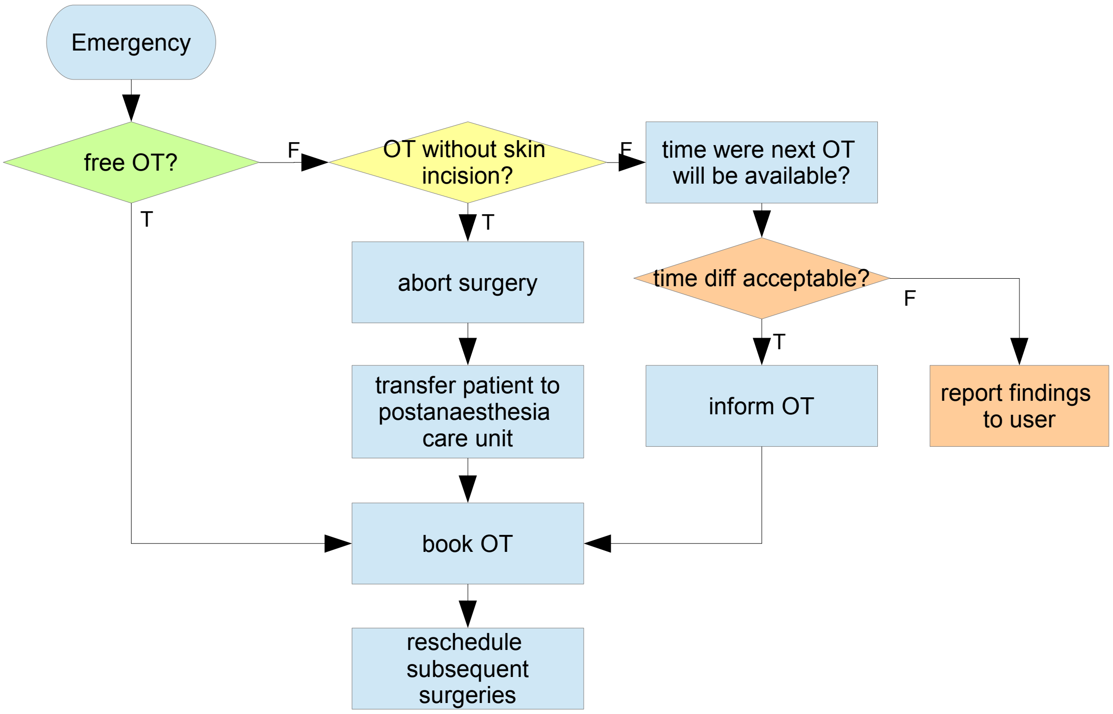

Emergency
free OT?
F
T
OT without skin incision?
F
T
time were next OT will be available?
abort surgery
time diff acceptable?
F
T
transfer patient to postanaesthesia care unit
inform OT
report findings to user
book OT
reschedule subsequent surgeries

Figure 4: Emergency scheduling flow chart

1.4 Checklists - Patient safety

The WHO initiated the "Safe Surgery Saves Lives" initiative in order to reduce the number of surgical deaths around the world. They published a checklist that divides the operation into three phases:

- Sign-In (before induction of anaesthesia)
- Time-Out (before skin incision)
- Sign-Out (before patient leaves operating room)

In each step, several attributes must be verified in order to proceed. (e.g. Sign-in: Does the patient have an allergy?) These checks are implemented as mandatory steps within the module, but alterations to this list should be possible (e.g. add a new check) [World Alliance for Patient Safety, 2008]

1.5 Development Process

In this section, I would like to demonstrate how I want to achieve the goal, to deliver a module, ready for production that contains the most valuable functions.

Therefore, I would like to apply principles from the Extreme Programming (XP) development process.

- Test driven development
- Simple design
- Incremental change
- Iterations

Unfortunately, it won't be possible to implement all desired functions at once. Hence, it is very important to select the most vital features and implement them first. According to XP, I will split the entire GSoC time into several iterations. In the first step, I will try to estimate the required development time of each Story Card. A story card describes the desired functionality from the user perspective. At the beginning of each iteration, my mentor (and I) will prioritize them and pick the most crucial ones. The output after the completion of a story will be a running module with a growing set of functions. At the end of the iteration I will track the progress and calculate the project velocity. This will help me to improve my estimations for future stories and is the foundation for the planning of the next period. This leads to a constant refinement and adaption of the project plan.

As a consequence, unforeseen changes and delays won't result in a complete project failure, but in fewer stories being implemented. Furthermore, this should help to quickly adapt to changing requirements along the project. Test driven development as well as constant refactoring will ensure high code quality. Finally, it ensures that no time is wasted on functionality that won't ever be used. [Beck, 1999]

1.6 User Stories

As already mentioned in the last chapter, user stories describe desired functionality that should be implemented from the users' view. They are estimated by the developer and handed back for prioritization. User stories are normally written in the form: "As <user_ role> I want <functionality _ xy> ." Please see the following examples:

essential

- As user I want an install able module
- As user I want a home screen linking to the basic functions
- As user I want to create a new surgery record
- As user I want to open en existing surgery record
- As user I want an overview inside the surgery record, viewing basic patient and surgery data
- As user I want to view scheduled surgeries
- As user I want to schedule a surgery
- As user I want to schedule an emergency
- As physician I want a section inside the surgery record that shows important patient data.
- As physician I want to enter important patient data (allergies, long-term medication, pre-existing illnesses)
- ...

valueable

- As user I want the system to automatically propose a scheduling
- As user I want to view the scheduled surgeries on a weekly basis
- As user I want to execute Reports
- As user I want a Report that shows the efficiency of the OTs
- As Anesthesiologist I want to automatically record vital signs during the surgery
- ...

nice to have

- As user I want to filter surgeries within the scheduling view by OT-Name, Surgeon and procedure
- As physician I want discharge letter templates that automatically fills in stored data.
- ...

1.7 Road Map

I would like to use an iteration length of two weeks. At the beginning, the most relevant stories will be selected in a meeting with my mentor. The following Road-Map describes a potential sequence for the development iterations.

Community Bonding period – Exploration Period
Begin: 04/21/14 – End: 05/16/14 – Duration: 20 days

- discuss project proposal with mentor
- finish writing all user stories and enter them into JIRA
- do an initial ranking with my mentor
- Further familiarize myself with Spring and Hibernate
- Do experiments
- study literature on scheduling
- determine which functionality is already implemented

First Coding Period
Begin: 05/19/14 – End: 06/27/14 – Duration: 27 days

- Development Iteration # 1 – Basics
- Development Iteration # 2 – Surgical record (General)
- Development Iteration # 3 – Scheduling

Second Coding Period
Begin: 06/30/14 – End: 08/18/14 – Duration: 34 days

- Development Iteration # 4 – Surgical record (intraoperative)
- Development Iteration # 5 – Surgical record (Patient)
- Development Iteration # 6 – Reports
- Last week – Finalize: Improve UI, Refactor code, write additional tests and documentation

2 Personal Information

2.1 Why I am the right person for this task

Since I have read about this project, I am really enthusiastic about it. Although, there are many interesting projects available for GSoC, none of them generated passion like this one.

I studied medicine for two years and worked as a medical clerk in the University Hospital of Graz. I spent two weeks in the department of radiology followed by two weeks in the operation theater assisting the surgical team during orthopedic interventions. As a result, I have knowledge about important processes within a surgical department.

At the moment, I am doing my masters degree in Software Development and Business Management at Graz, University of Technology (GPA: 3.85/4.0). For my bachelor thesis I developed a "waiting time management module for physicians' private practices" that seamlessly integrates with one of the market leading practice management software's in Austria.

I did two internships within the health sector and gained experience in dealing with large systems. I substantially reduced pending bugs and change requests for reports in an EMR-System tailored for nursing homes. In 2012 I interned at the global company "Computer Sciences Corporation" (CSC) and was responsible for a small project. I wrote novel documentation for a large software system in order to close the knowledge gap between software engineers and non-technical personnel.

Furthermore, I attended an online course of the University Duisburg-Essen (Germany) in "Integrated Business Processes with SAP ERP" [1]

Additionally, I plan to credit GSoC for a project that I have to do for my studies. This means that I will also be supervised from a professor at my university if I have questions that are not particularly related to the internals of OpenMRS.

To sum it up, I think this project is ideally suited for my educational background.

[1] http://www.erp4students.co.uk/live/Course-Integrated-Business-Processes-SAP-ERP/integrated-processes.aspx

2.2 Software Development Experience

2.2.1 Projects

- Verification and Testing
  Implemented Advanced Eraser and Locktree algorithm – Java
- Security Aspects in Software Development
  Searched for and fixed security vulnerabilities in a web application – Java, Javascript, SQL
  Implemented large parts of a BigInteger Library – C
  Tiny VM: Implemented source code tokenizing, Bytecode loading, Stack, ... – C
- Summer Internship
  Substantially reduced pending bugs and change requests for reports in an EMR-System using Oracle and MSSQL databases
- Bachelor thesis
  Developed "waiting time management module" for physicians' private practices – Java Servlets, SQL, JavaScript
- Web Science and Web Technology
  Implemented Twitters TunkRank algorithm in a Map/Reduce Environment – Python
- Design and Development of Large Systems:
  Developed small Raytracer – C++
- Operating Systems
  Implemented Multithreading, Filesystem access, Locks, Memory Swapping and Memory mapped file I/O for an educational operating system called SWEB – C++
- Software Development in Distributed Systems
  Developed Messaging System - Cell phones transmit messages to LAN Clients; this was done over Bluetooth through a gateway server – Java
- StartUp - coupled
  Created embeddable instant messaging solution for social networks using Java, JavaScript and Actionscript

2.2.2 List of technologies, programming languages and tools...

...at least used once during a project – listed in no particular order

Languages	Libraries	Technologies
Java	JUnit, CUnit, httpUnit	Eclipse
JavaScript	Javassit (bytecode instrumentation)	Git
SQL	IAIK Java JCE provider (cryptography)	MySQL, MSSQL, Oracle
C/C++	JFreeChart	Tomcat
Python	Asterisk-java (connector)	SAP
Matlab / Octave	JNI (java native interface)	Valgrind
PHP	JML (java modeling language)	Asterisk Telephony Server
	Klee (symbolic execution)	Hadoop
	Xerces-J (XML-Parser)	
	Jmockit (testing)	
	OpenCV (computer vision library)	
	Bluecove (Java Bluetooth library)	

2.3 Contributions and interaction with the community

Ticket	Pull-Request
TRUNK-4166	https://github.com/openmrs/openmrs-core/pull/725
SYNC-301	work in progress – see JIRA Ticket Conversation

Channel	
Mailing-List	lukas.breitwieser@student.tugraz.at
IRC	lbat
Developer Forum	3/06/2014
	3/20/2014 (last 20 minutes)
Personal Wiki Space	https://wiki.openmrs.org/pages/viewpage.action?pageId=60069261
	https://wiki.openmrs.org/display/ lbat/GSoC+Operation+Theater+ Module

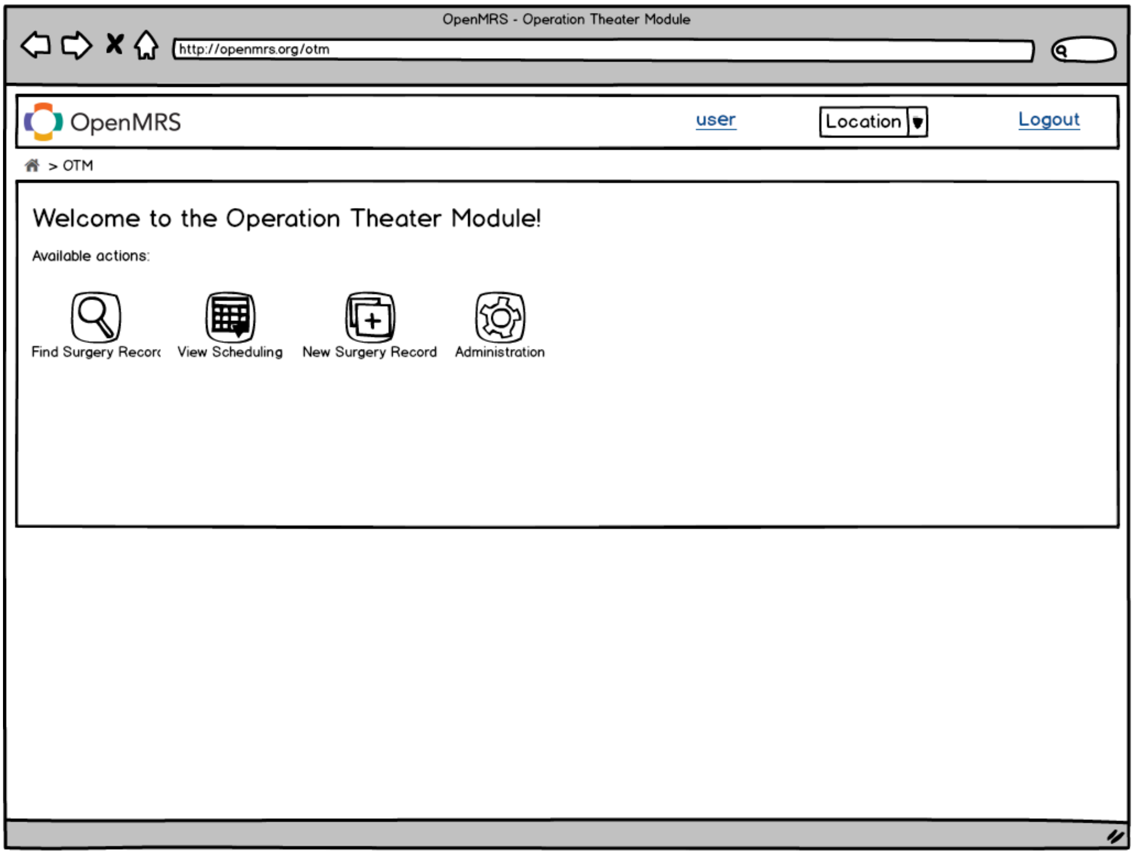
I introduced myself on the mailing list and discussed a few ideas about the OT-Module. I also published my initial thoughts on the wiki for discussion across the community. On IRC I was following conversations and asking for (ticket related) questions.

2.4 Other responsibilities/commitments

During the Google Summer of Code period I plan to work full time on this project. Along the way I am only attending two lectures which require less than five hours a week. In order to work solely on this module, I have to finish all my other responsibilities before GSoC begins. As a result I have been very busy and have not been able to solve as many tickets as I would like to do. Nevertheless, I hope I could demonstrate my motivation and enthusiasm.

3 Appendix

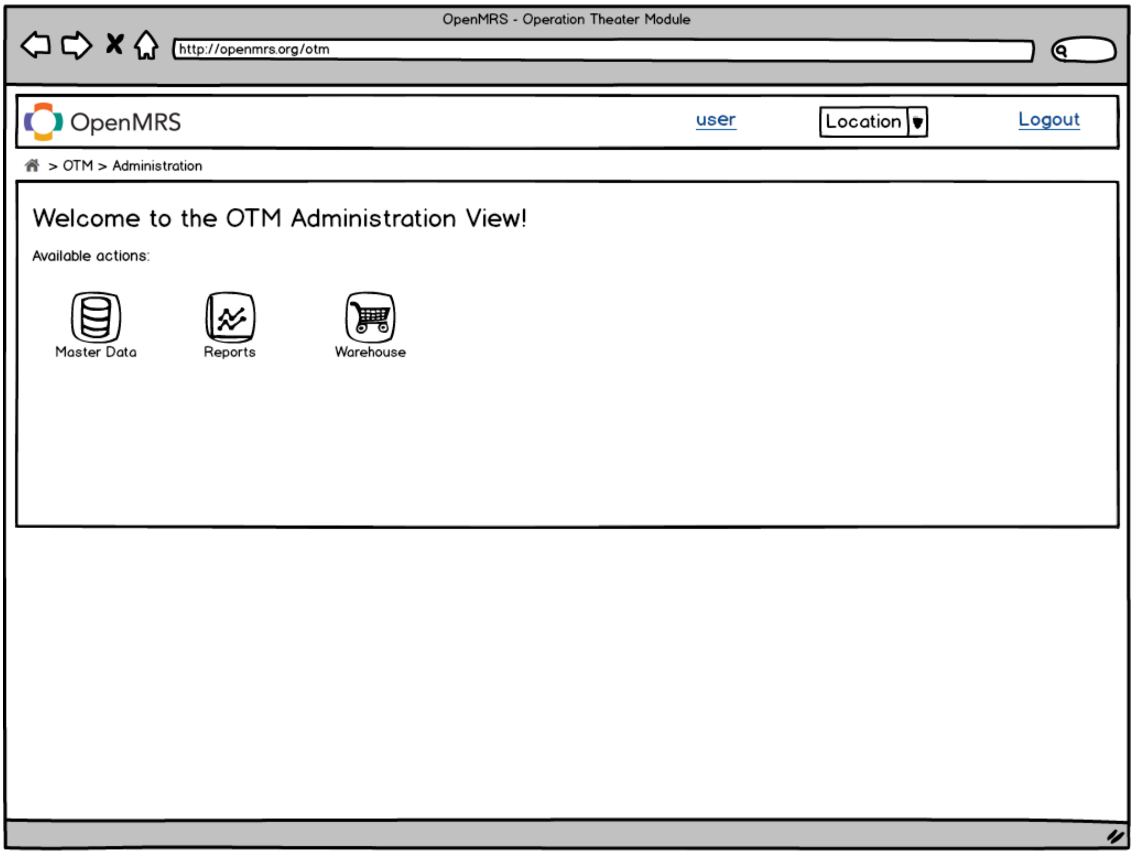
3.1 UI-Views



Welcome to the Operation Theater Module!

Available actions:

Figure 5: Home page



Welcome to the OTM Administration View!

Available actions:

Figure 6: Administration page

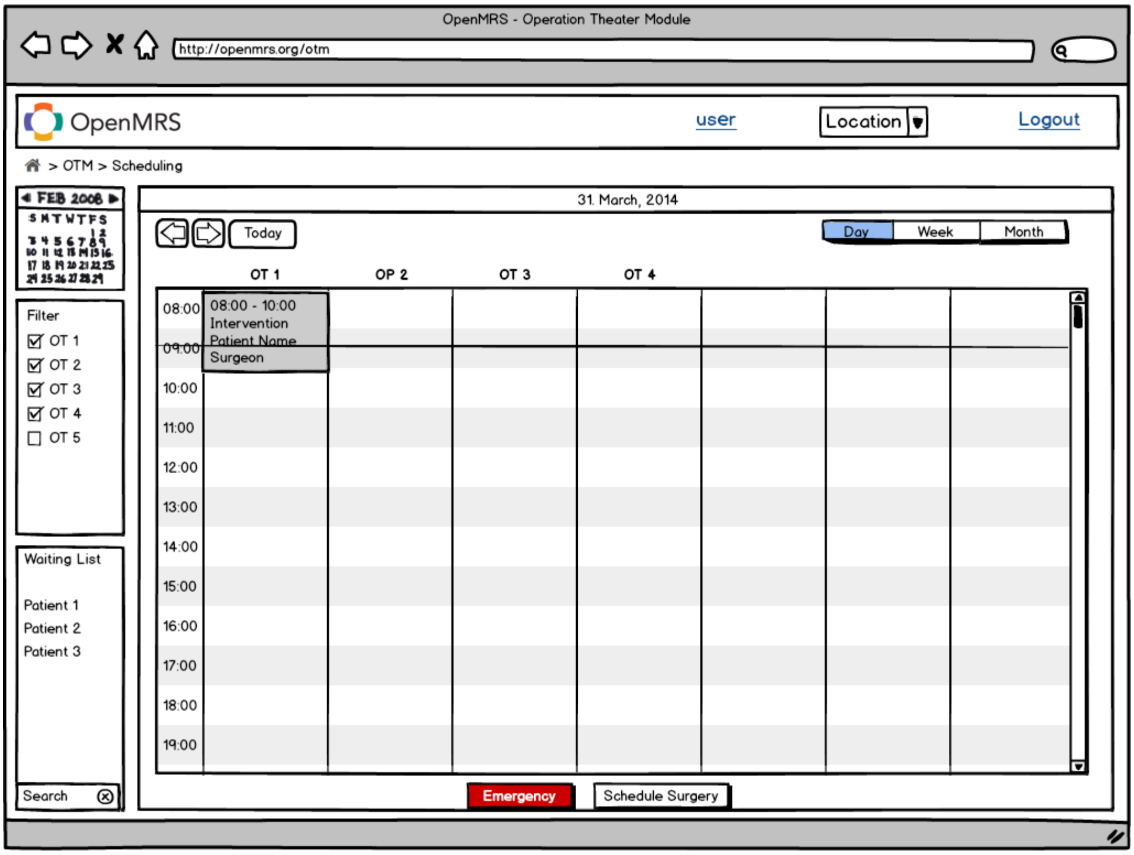
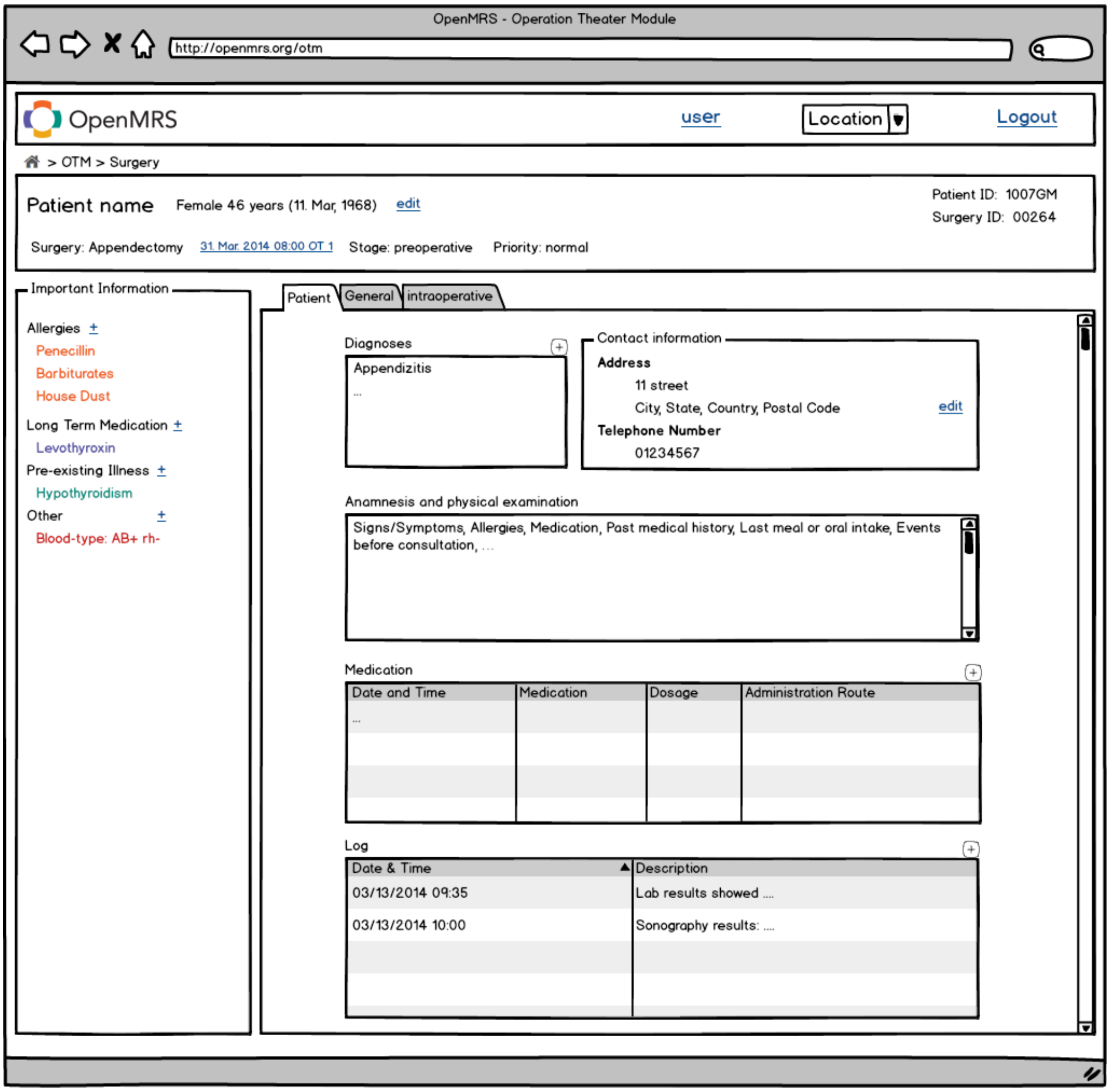


Figure 7: Scheduling View



Patient name Female 46 years (11. Mar, 1968)

Patient ID: 1007GM

Surgery ID: 00264

Surgery: Appendectomy 31 Mar 2014 08:00 OT 1 Stage: preoperative Priority: normal

Important Information

Allergies +

Penecillin

Barbiturates

House Dust

Long Term Medication +

Levothyroxin

Pre-existing Illness +

Hypothyroidism

Other +

Blood-type: AB+ rh-

Diagnoses

Appendizitis

...

Contact information

Address

11 street

City, State, Country, Postal Code

Telephone Number

01234567

Anamnesis and physical examination

Signs/Symptoms, Allergies, Medication, Past medical history, Last meal or oral intake, Events before consultation, ...

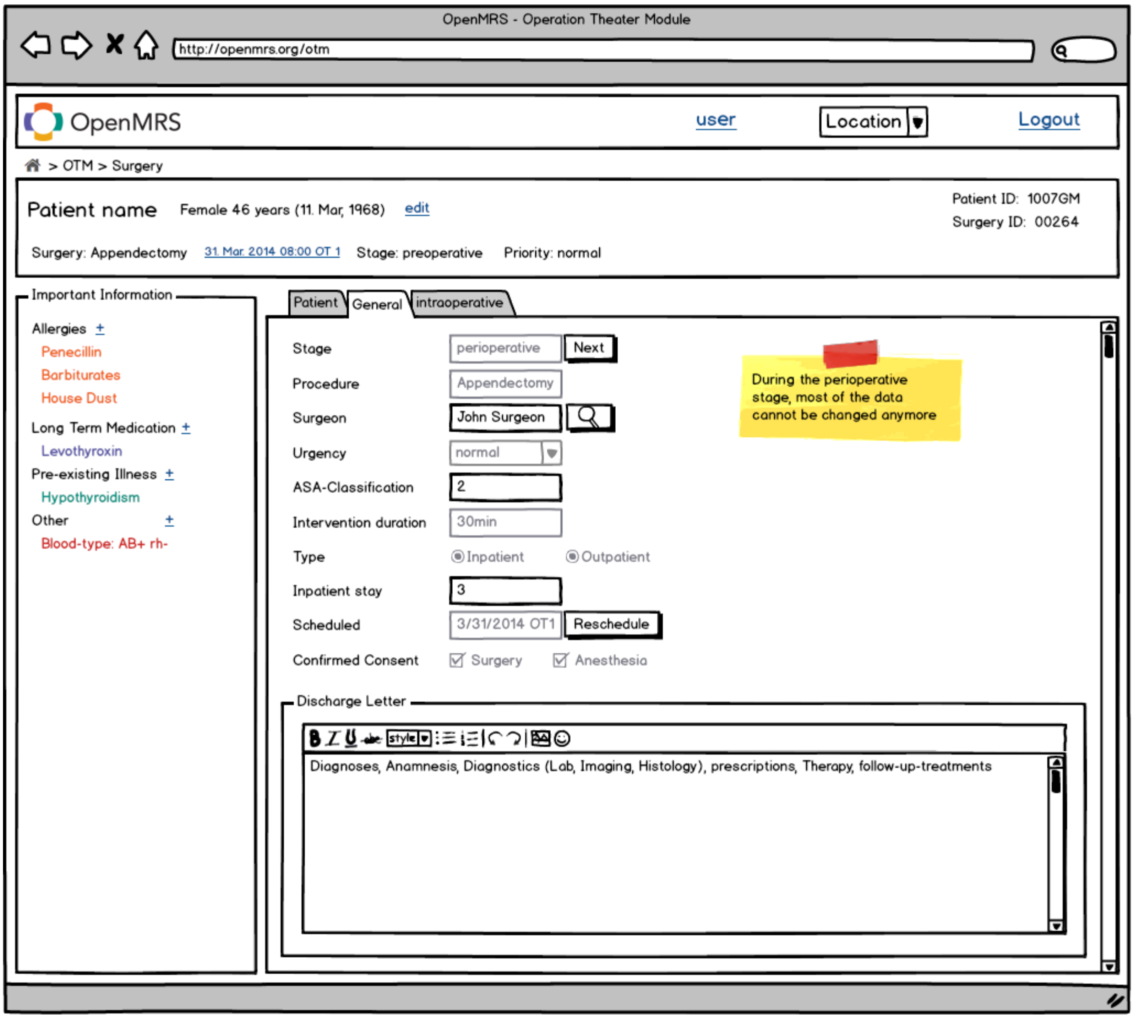
Medication

Date and Time	Medication	Dosage	Administration Route
...			

Log

Date & Time ▲	Description
03/13/2014 09:35	Lab results showed ...
03/13/2014 10:00	Sonography results: ...

Figure 8: Surgical Record - Patient Tab



OpenMRS - Operation Theater Module

http://openmrs.org/otm

OpenMRS user Location Logout

> OTM > Surgery

Patient name Female 46 years (11. Mar, 1968) edit

Patient ID: 1007GM
Surgery ID: 00264

Surgery: Appendectomy 31. Mar 2014 08:00 OT 1 Stage: preoperative Priority: normal

Important Information

Allergies +
Penecillin
Barbiturates
House Dust

Long Term Medication +
Levothyroxin

Pre-existing Illness +
Hypothyroidism

Other +
Blood-type: AB+ rh-

Patient | General | intraoperative

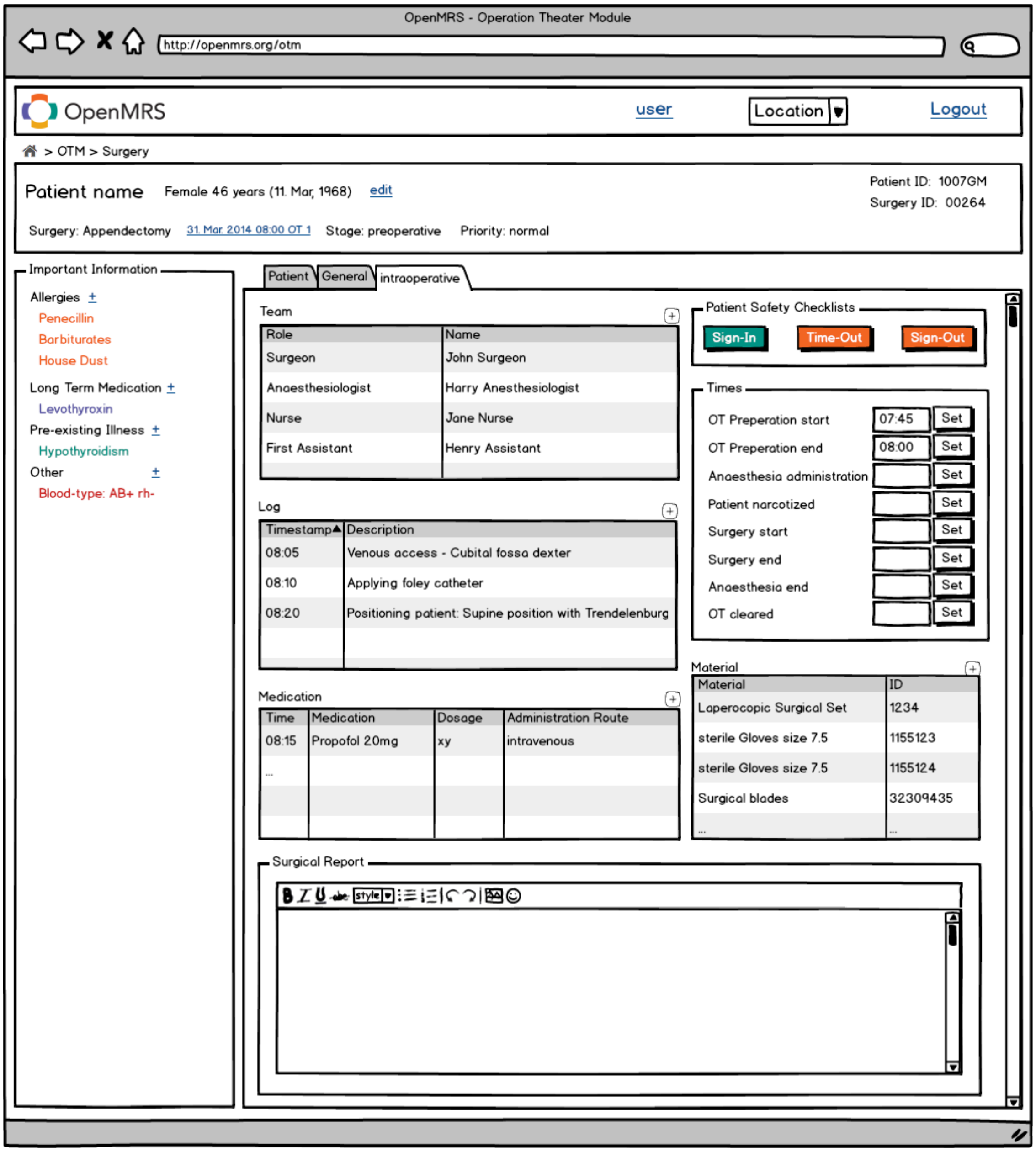
Stage	perioperative [Next]
Procedure	Appendectomy
Surgeon	John Surgeon
Urgency	normal
ASA-Classification	2
Intervention duration	30min
Type	● Inpatient ● Outpatient
Inpatient stay	3
Scheduled	3/31/2014 OT1 [Reschedule]
Confirmed Consent	☑ Surgery ☑ Anesthesia

During the perioperative stage, most of the data cannot be changed anymore

Discharge Letter

Diagnoses, Anamnesis, Diagnostics (Lab, Imaging, Histology), prescriptions, Therapy, follow-up-treatments

Figure 9: Surgical Record - General Tab



> OTM > Surgery

Patient name Female 46 years (11. Mar, 1968) edit

Patient ID: 1007GM
Surgery ID: 00264

Surgery: Appendectomy 31. Mar. 2014 08:00 OT 1 Stage: preoperative Priority: normal

Important Information

Allergies +
Penecillin
Barbiturates
House Dust

Long Term Medication +
Levothyroxin

Pre-existing Illness +
Hypothyroidism

Other +
Blood-type: AB+ rh-

Team

Role	Name
Surgeon	John Surgeon
Anaesthesiologist	Harry Anesthesiologist
Nurse	Jane Nurse
First Assistant	Henry Assistant

Log

Timestamp▲	Description
08:05	Venous access - Cubital fossa dexter
08:10	Applying foley catheter
08:20	Positioning patient: Supine position with Trendelenburg

Medication

Time	Medication	Dosage	Administration Route
08:15	Propofol 20mg	xy	intravenous
...			

Patient Safety Checklists

Times

OT Preperation start	07:45	Set
OT Preperation end	08:00	Set
Anaesthesia administration		Set
Patient narcotized		Set
Surgery start		Set
Surgery end		Set
Anaesthesia end		Set
OT cleared		Set

Material

Material	ID
Laperocopic Surgical Set	1234
sterile Gloves size 7.5	1155123
sterile Gloves size 7.5	1155124
Surgical blades	32309435
...	...

Surgical Report

Figure 10: Surgical Record - intraoperative Tab

References

Agnoletti et al. Operating room data management: improving efficiency and safety in a surgical block. *BMC Surgery*, 13, 2013.

Kent Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Longman, 1999.

Welk and Bauer. *OP-Management: praktisch und effizient*. Springer, 2006.

World Alliance for Patient Safety. WHO surgical safety checklist and implementation manual, 2008. URL `http://www.who.int/patientsafety/safesurgery/ss_checklist/en/`.