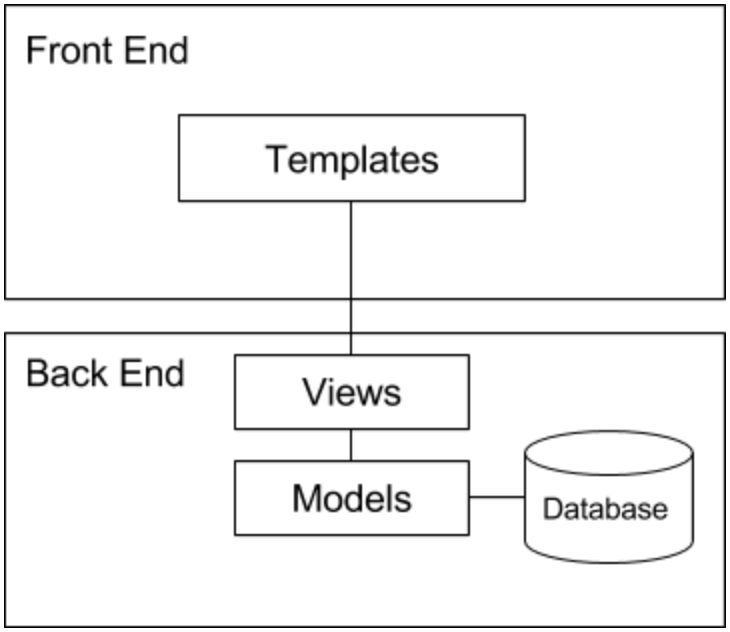
Product Design

Team Suites

Revision Number	*Revision Date*	*Summary of Changes*	*Author(s)*
0.1	*09/19/16*	*Created the first revision of each part of the document.*	*Software Development Team*
1.0	*10/3/16*	*Finalized the product design document for submitting with Release 1*	*Software Development Team*
2.0	10/15/16	*Revised the product design document for R2 Planning*	*Software Development Team*
2.1	12/3/16	*Revised the product design document for submitting with Release 2*	*Software Development Team*

Architectural Model

- This diagram represents the major subsystems of the product. Initially focus on the domain layer and its components before decomposing the user interface component. Note that a common interface allows both the GUI and a Command Line Interface to access the domain model in the same manner without regard to the type of presentation technique.



Components and Functions

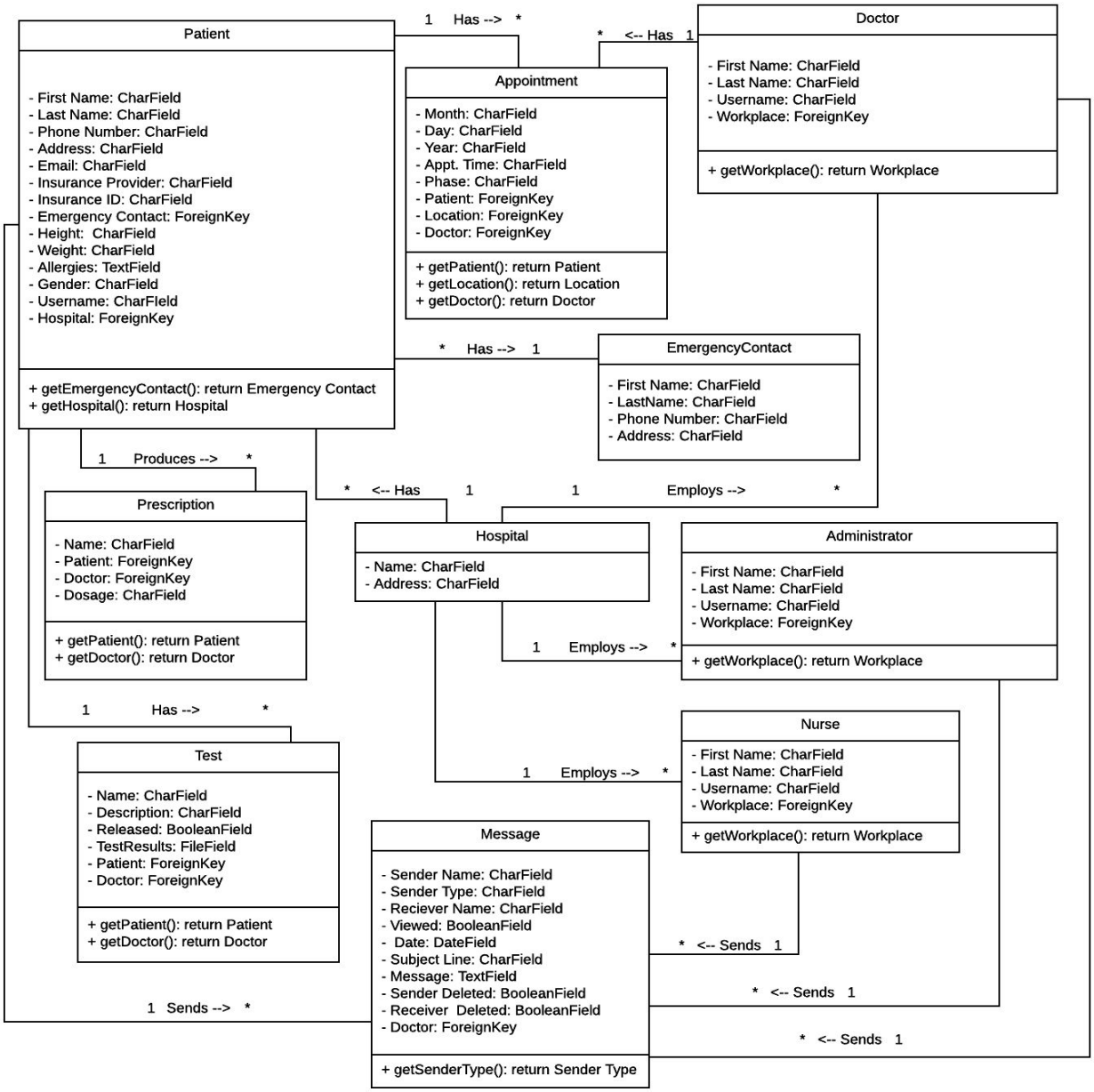
Patient	**Component state:** ● A Patient's personal, medical and insurance information **Component behavior:** ● Patients can update their own personal information ● Patients can request to create, update, and cancel appointments ● Patients can request to view their prescriptions ● Patients can request to view the calendar ● Patients can request to send messages and view and delete received messages

	● Patients can request to view released test results ● Patients can request to export tests and profile information
Nurse	**Component state:** ● A Nurse's name, username, and workplace **Component behavior:** ● Nurses can request to update patient information ● Nurses can request to create/update appointments ● Nurses can request to view prescriptions, patient information and the calendar ● Nurses can request to admit a patient
Test	**Component state:** ● All of the information for a given patient's test results (Name of test, Description of the test, Patient, Hospital, Doctor, and whether or not the text has been released) **Component behavior:** ● Test results can be released to Patients ● Test results can be created/updated by Doctors
Appointment	**Component state:** ● The patient who the appointment is for, the doctor that is meeting with the patient, as well as the date, time, and the location of the appointment **Component behavior:** ● Appointments can be created by Doctors and Patients ● Appointments can be updated by Patients, Doctors and Nurses

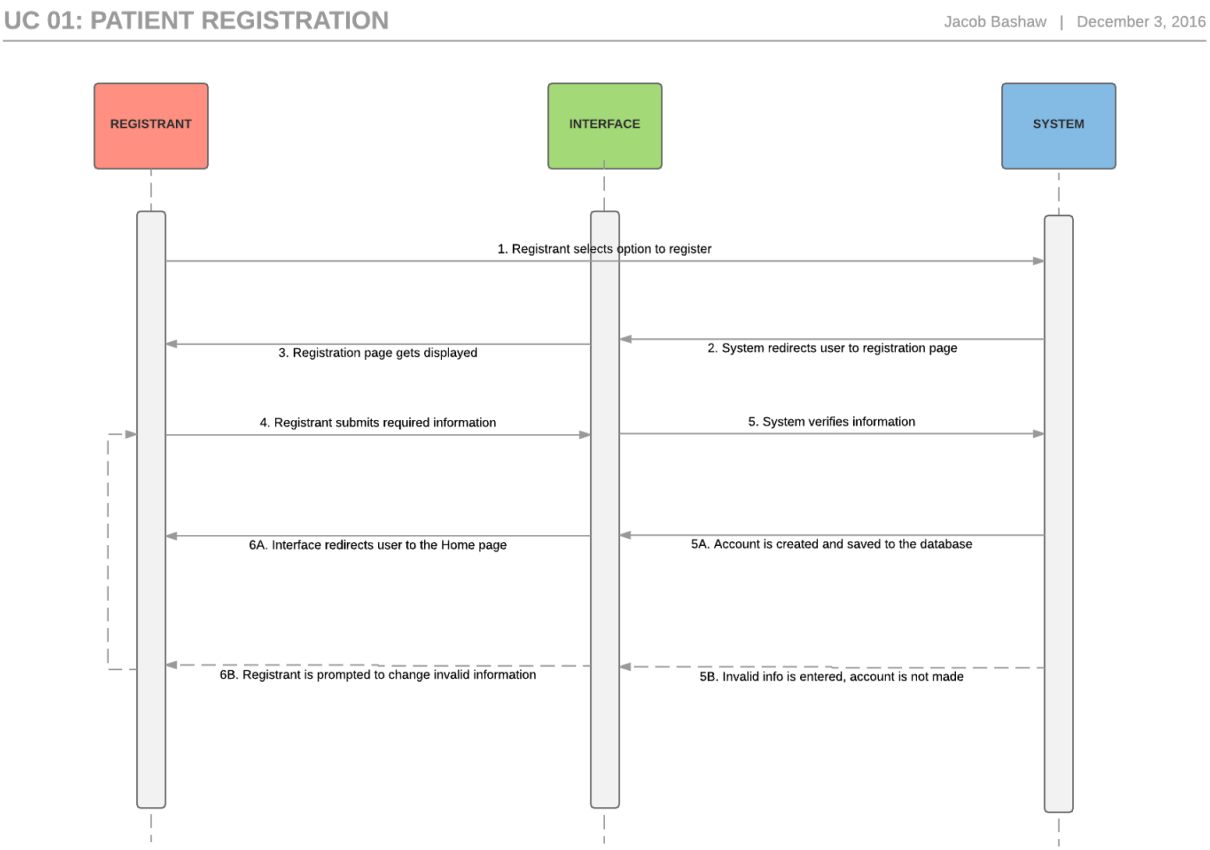
Doctor	**Component state:** ● A Doctor's personal information, workplace, and the appointments that they currently have **Component behavior:** ● Doctors can request to create/update appointments ● Doctors can request to update patient information ● Doctors can request to transfer patients ● Doctors can request to create/edit test results ● Doctors can request to create/remove prescriptions ● Doctors can request to release test results ● Doctors can request to admit/discharge a patient ● Doctors can request to view the calendar ● Doctors can request to upload test results
Administrator	**Component state:** ● An Administrator's personal information and workplace **Component behavior:** ● Administrators can request to add new workers (Nurses, Doctors, and Administrators) to their hospital(s) ● Administrators can request to view the logs and statistics for their hospital(s) ● Administrators can request for the transfer of a patient
Hospital	**Component state:** ● The information for the hospital (name, address) ● Lists of workers (Doctors, Nurses and Administrators) as well as Patients currently at the hospital **Component behavior:**

	● Hospitals can be used in creating appointments, and assigning workers a workplace
Prescription	**Component state:** ● A prescription name, amount and patient **Component behavior:** ● Prescriptions can be updated and created by Doctors

Class Diagram(s)

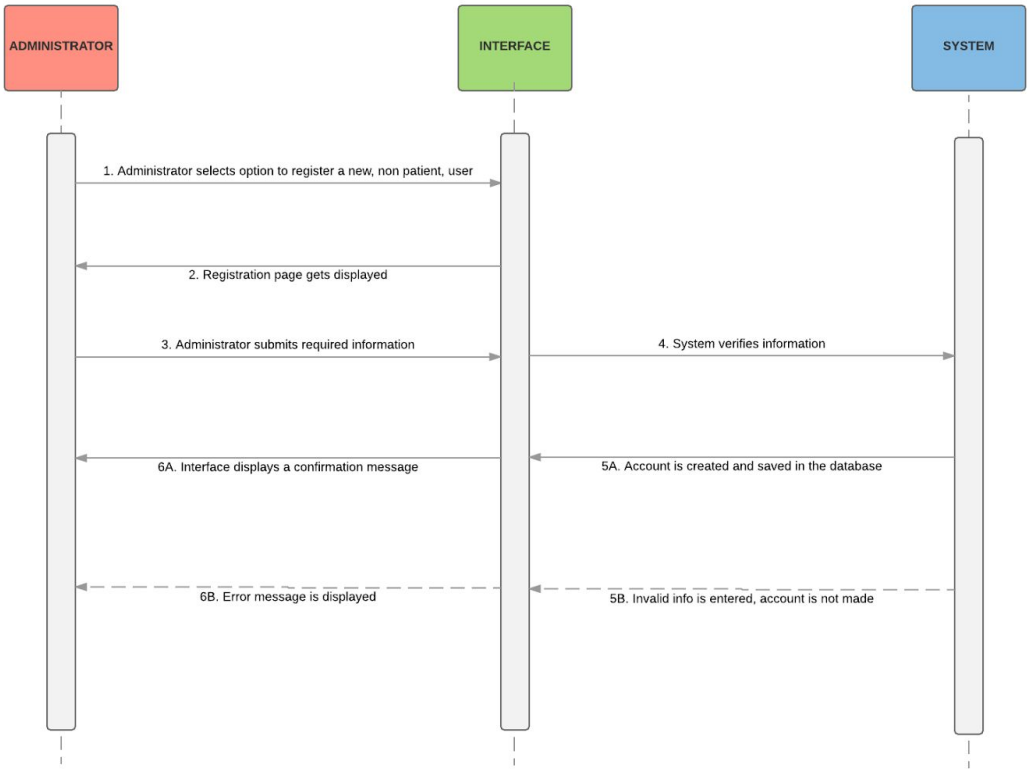


Sequence Diagram(s)



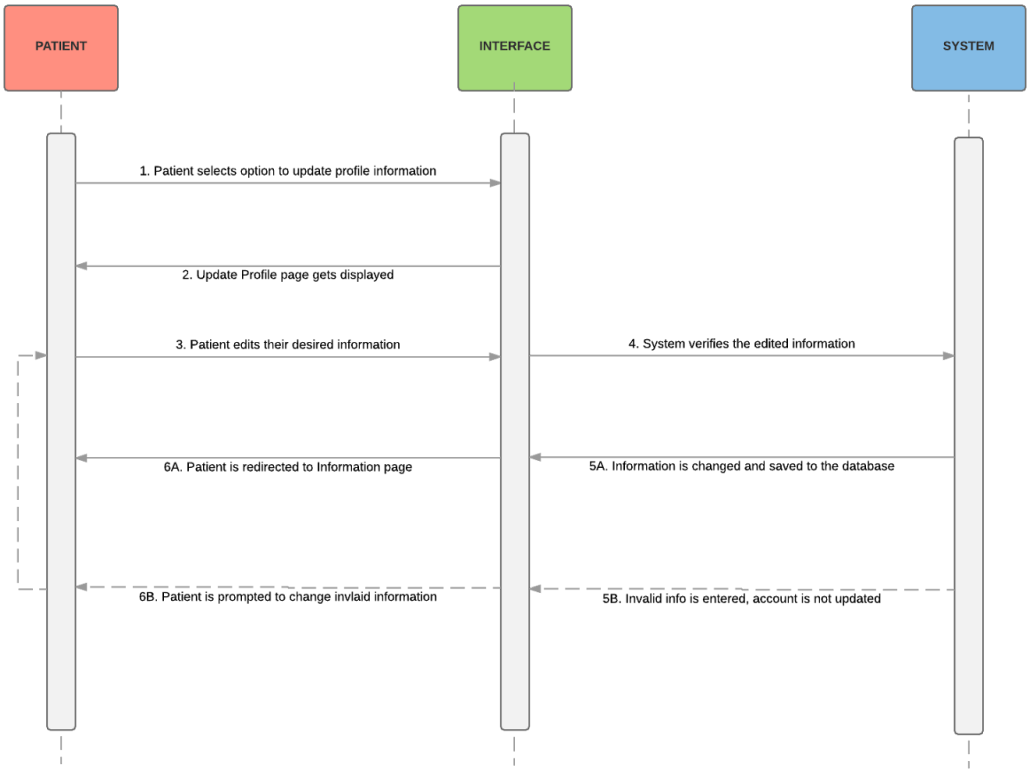
UC 02: ADMINISTRATOR REGISTRATION

dym5062 | October 17, 2016



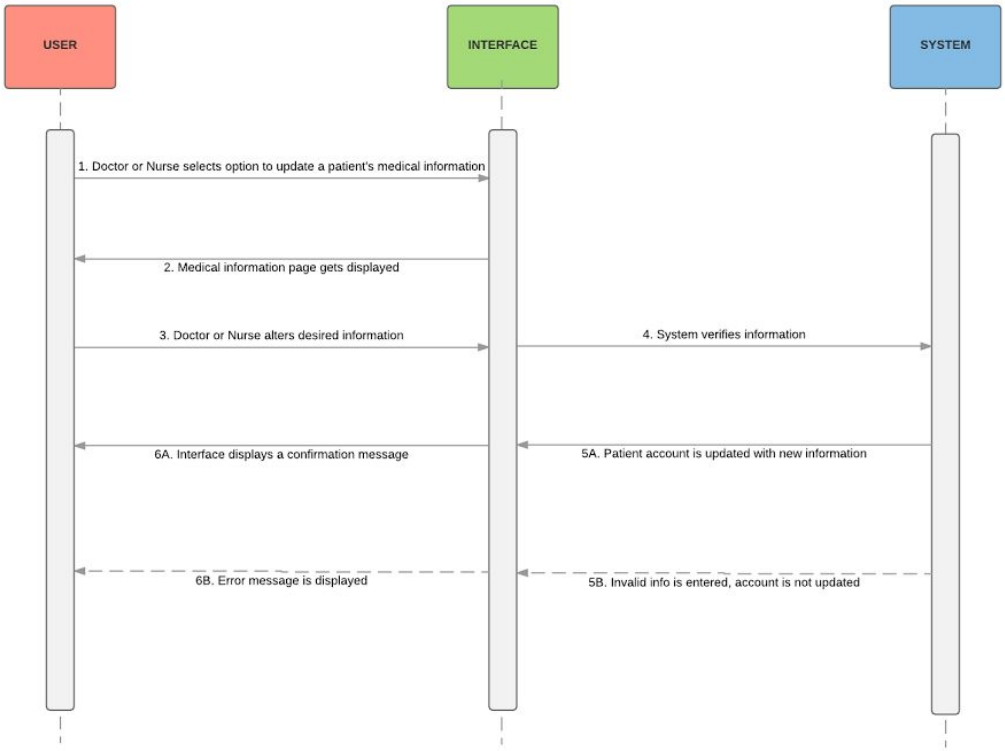
UC 03: UPDATE PATIENT PROFILE INFORMATION

Jacob Bashaw | December 3, 2016



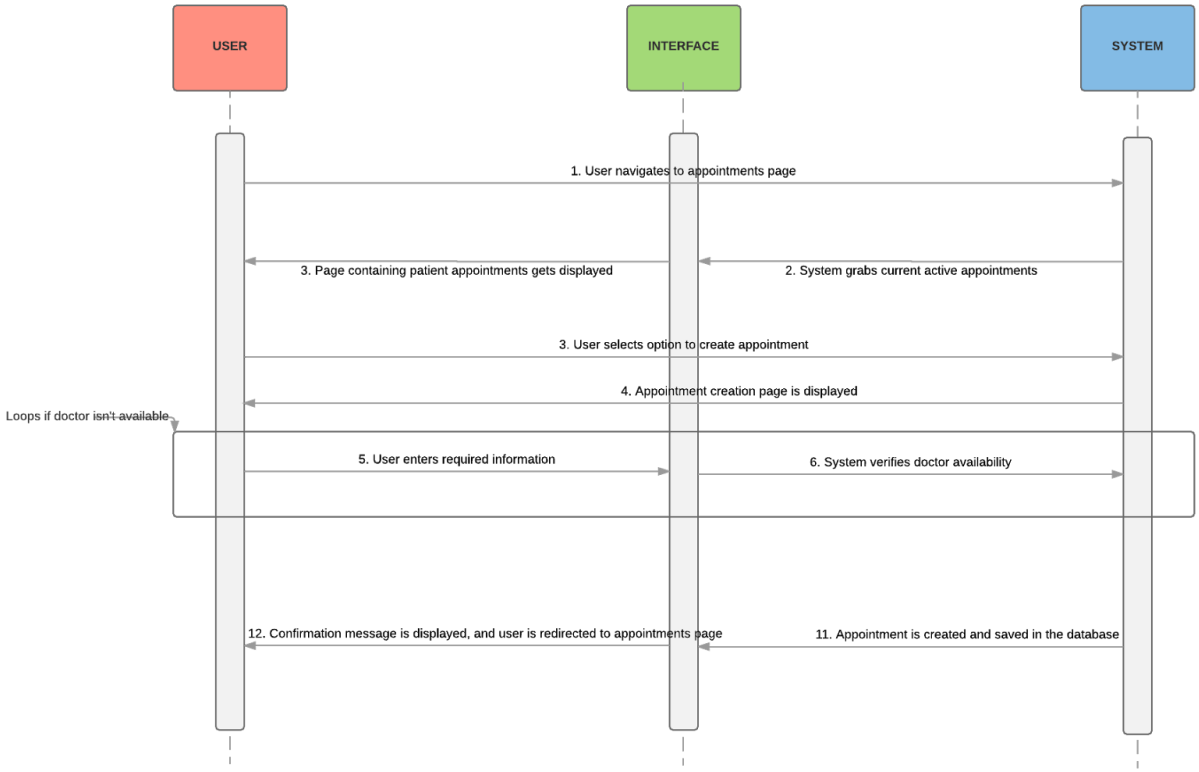
UC 04: UPDATE PATIENT MEDICAL INFORMATION

dym5062 | October 17, 2016



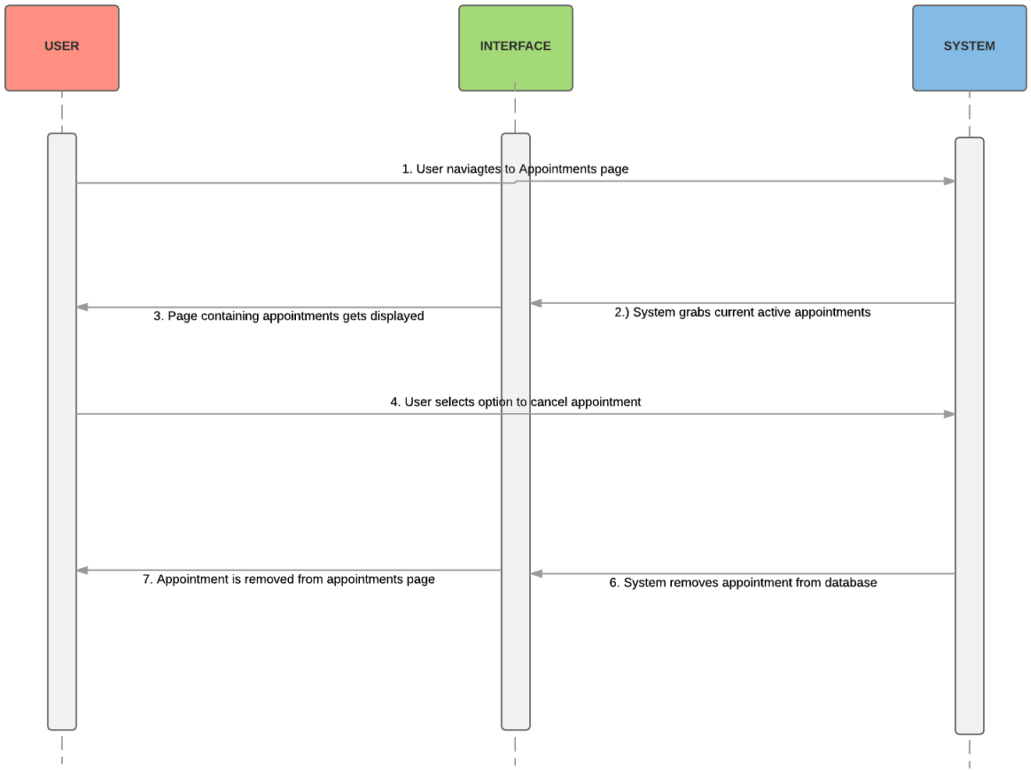
UC 06: CREATE/UPDATE PATIENT APPOINTMENT

Jacob Bashaw | December 3, 2016



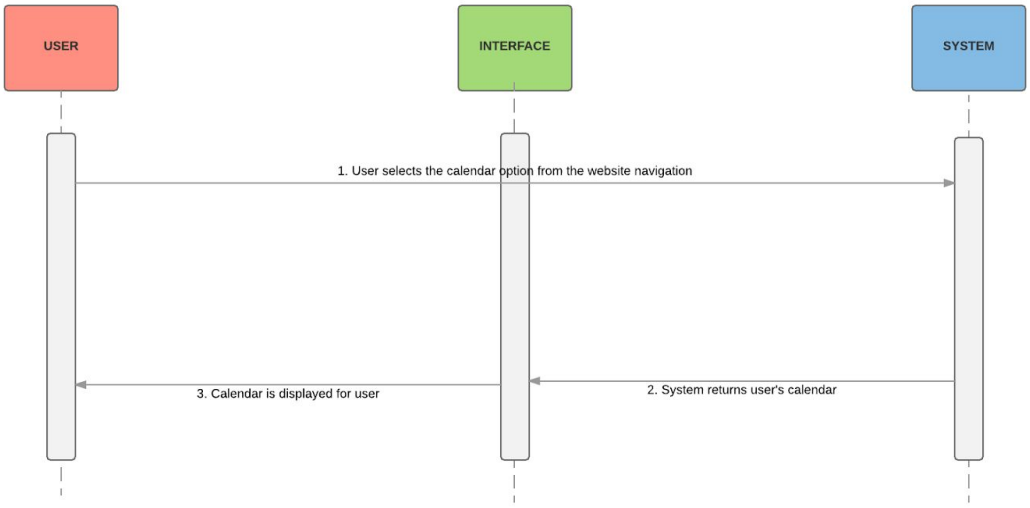
UC 07: CANCEL PATIENT APPOINTMENT

Jacob Bashaw | December 3, 2016



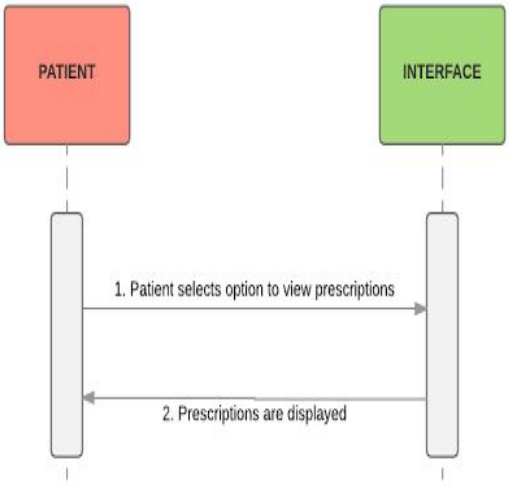
UC 08: APPOINTMENT CALENDAR

Jacob Bashaw | October 17, 2016



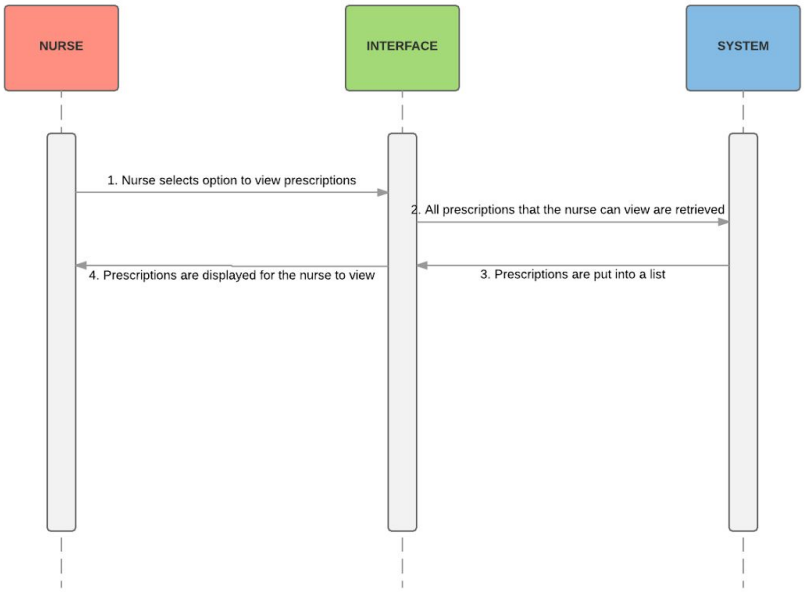
UC 09A: PATIENT VIEWING PRESCRIPTIONS

dym5062 | October 17, 2016



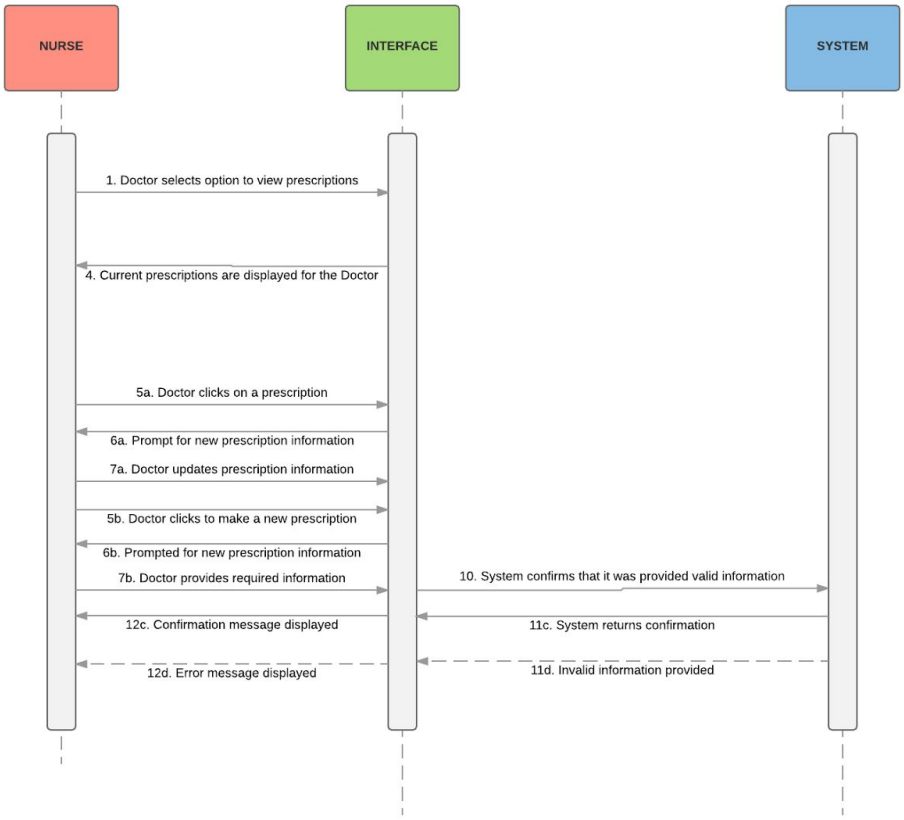
UC 09B: NURSE VIEWING PATIENT PRESCRIPTIONS

dym5062 | December 4, 2016



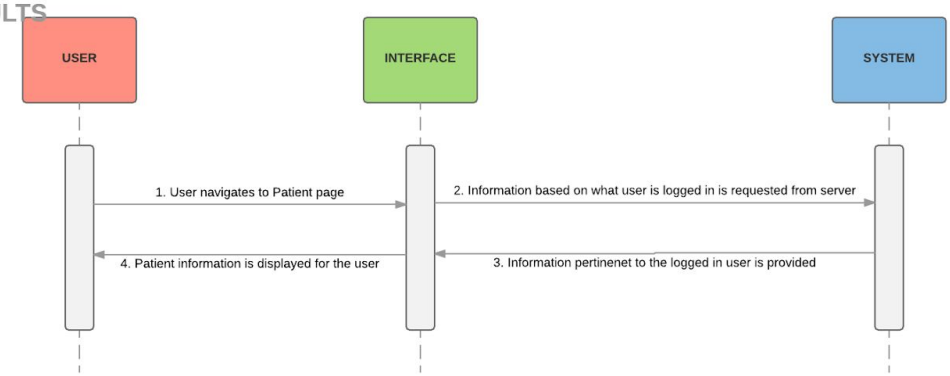
UC 09C: DOCTOR VIEWING AND EDITING PATIENT PRESCRIPTIONS

dym5062 | December 4, 2016



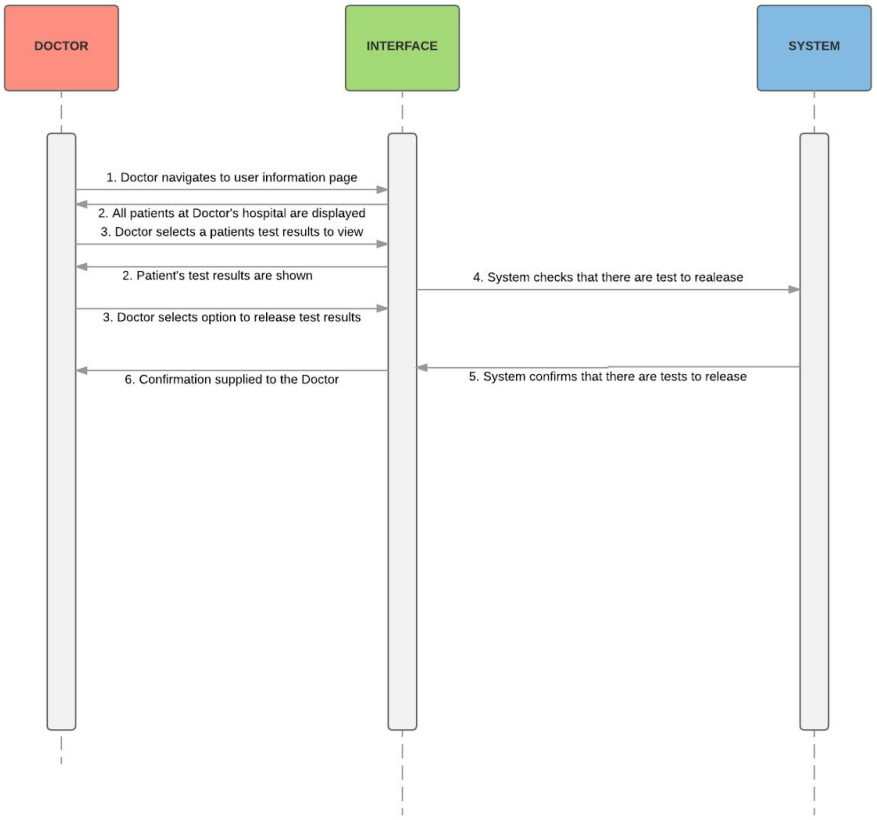
UC 10: VIEWING PATIENT MEDICAL INFORMATION, PRESCRIPTIONS AND TEST RESULTS

dym5062 | October 17, 2016



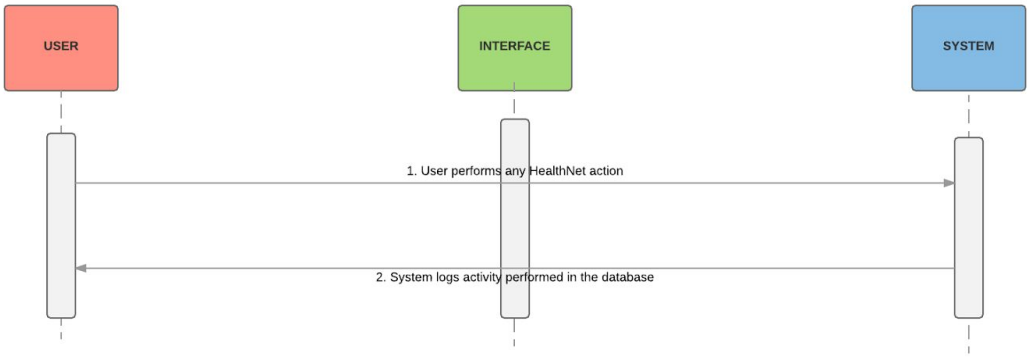
UC 11: RELEASE TEST RESULTS

dym5062 | December 4, 2016



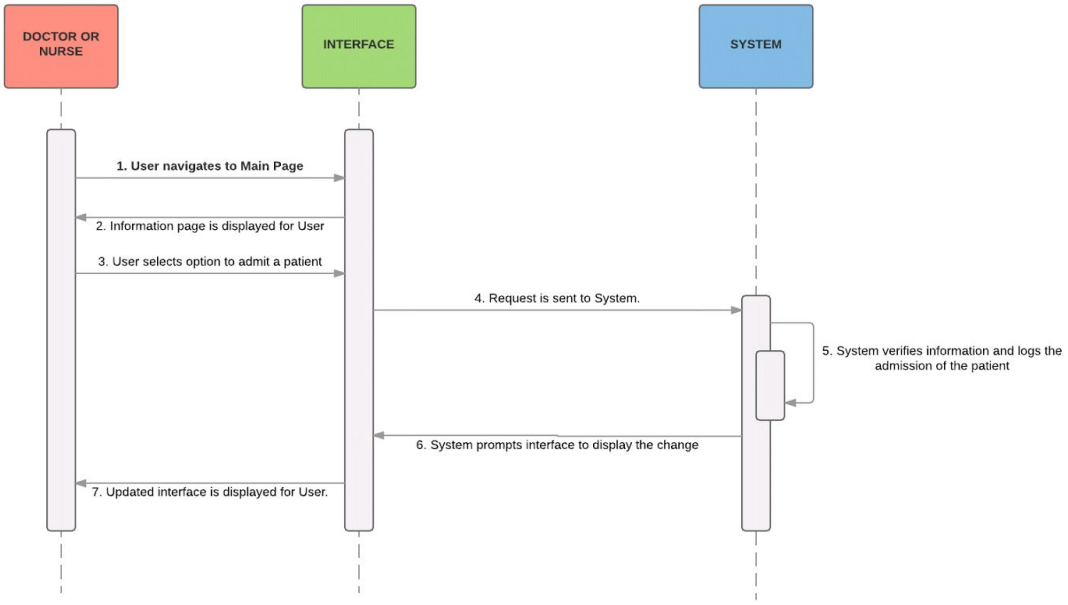
UC 12: LOGGING SYSTEM ACTIVITY

Jacob Bashaw | October 17, 2016



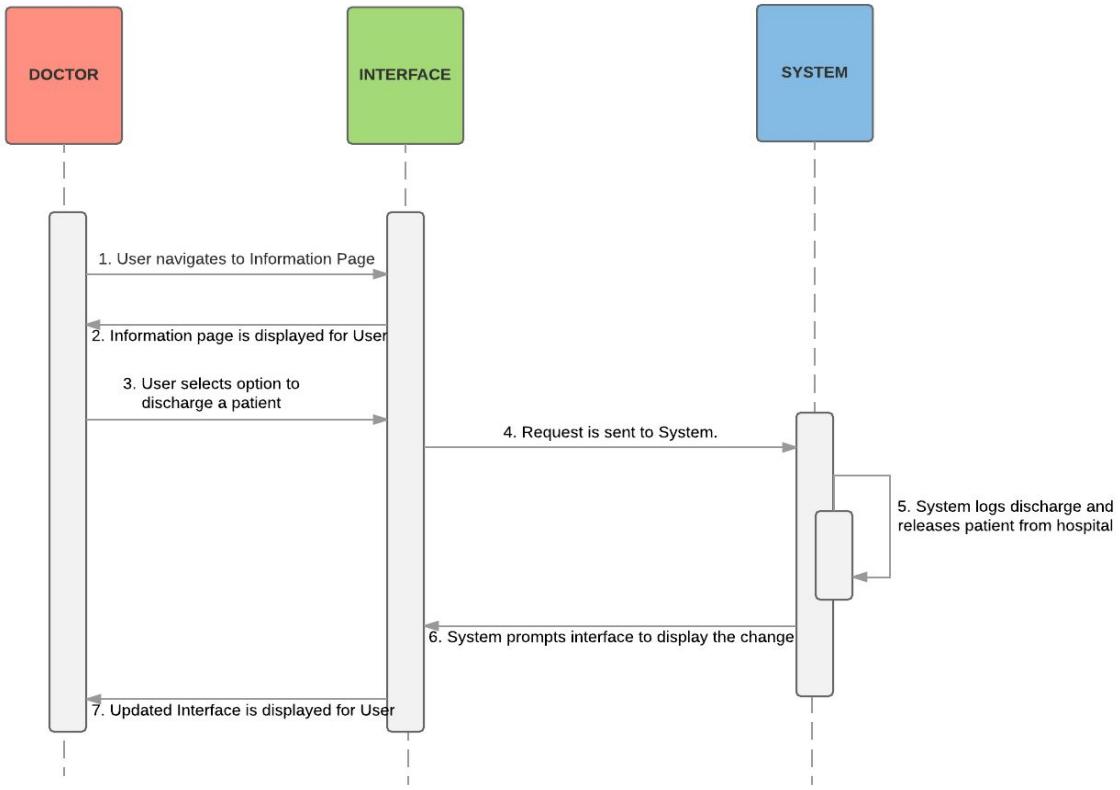
UC 13A: ADMISSION TO HOSPITAL

Michael Gilmour | December, 5th 2016



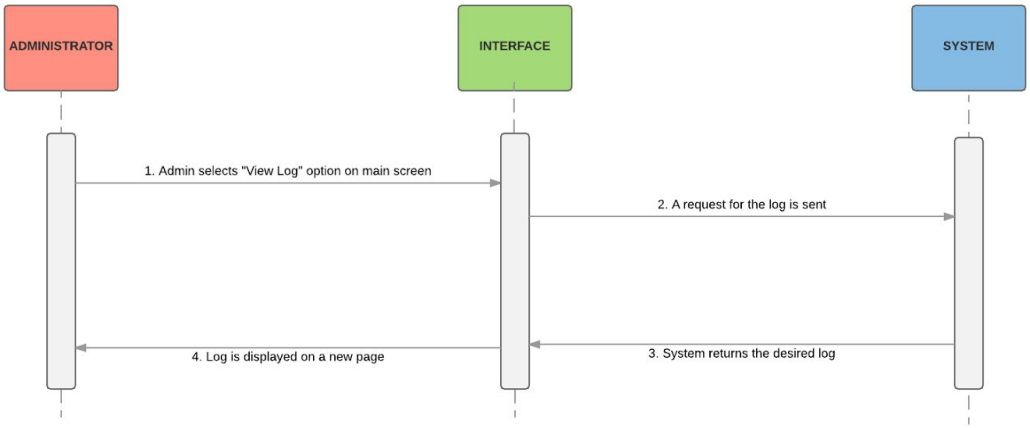
UC 13B: DISCHARGE FROM HOSPITAL

Michael Gilmour | December 6, 2016



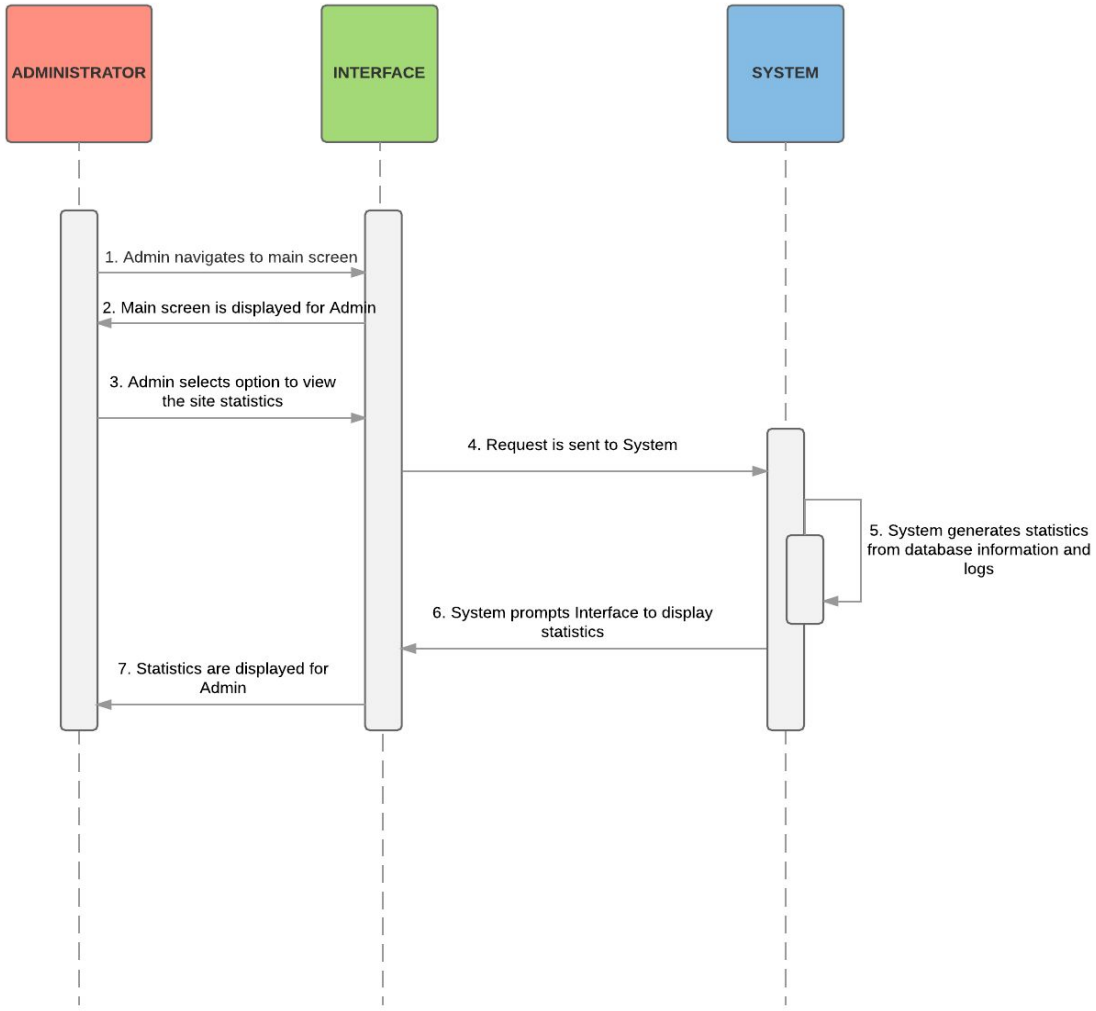
UC 14: VIEWING ACTIVITY LOG

Jacob Bashaw | October 17, 2016



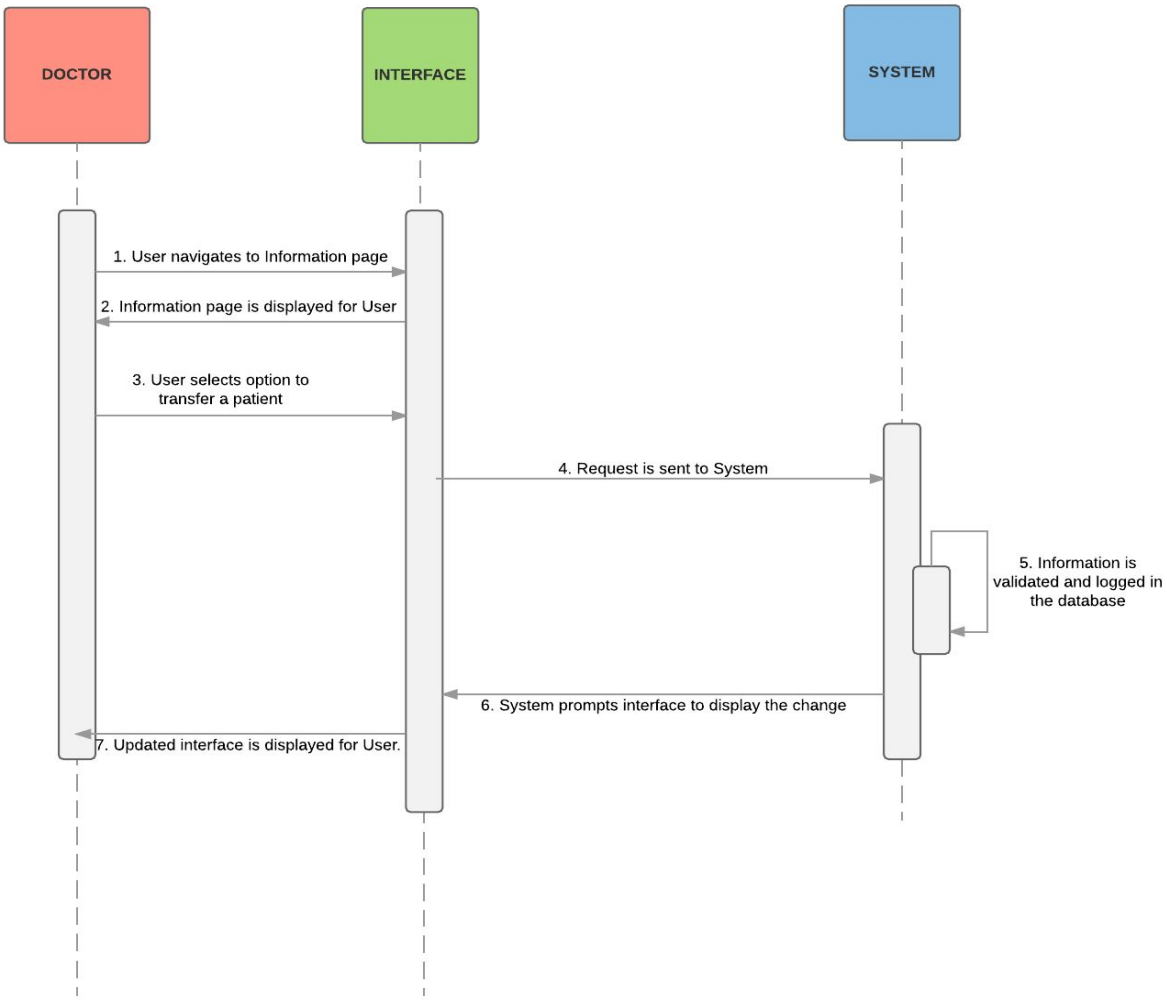
UC 15: VIEWING SYSTEM STATISTICS

Michael Gilmour | October 18, 2016



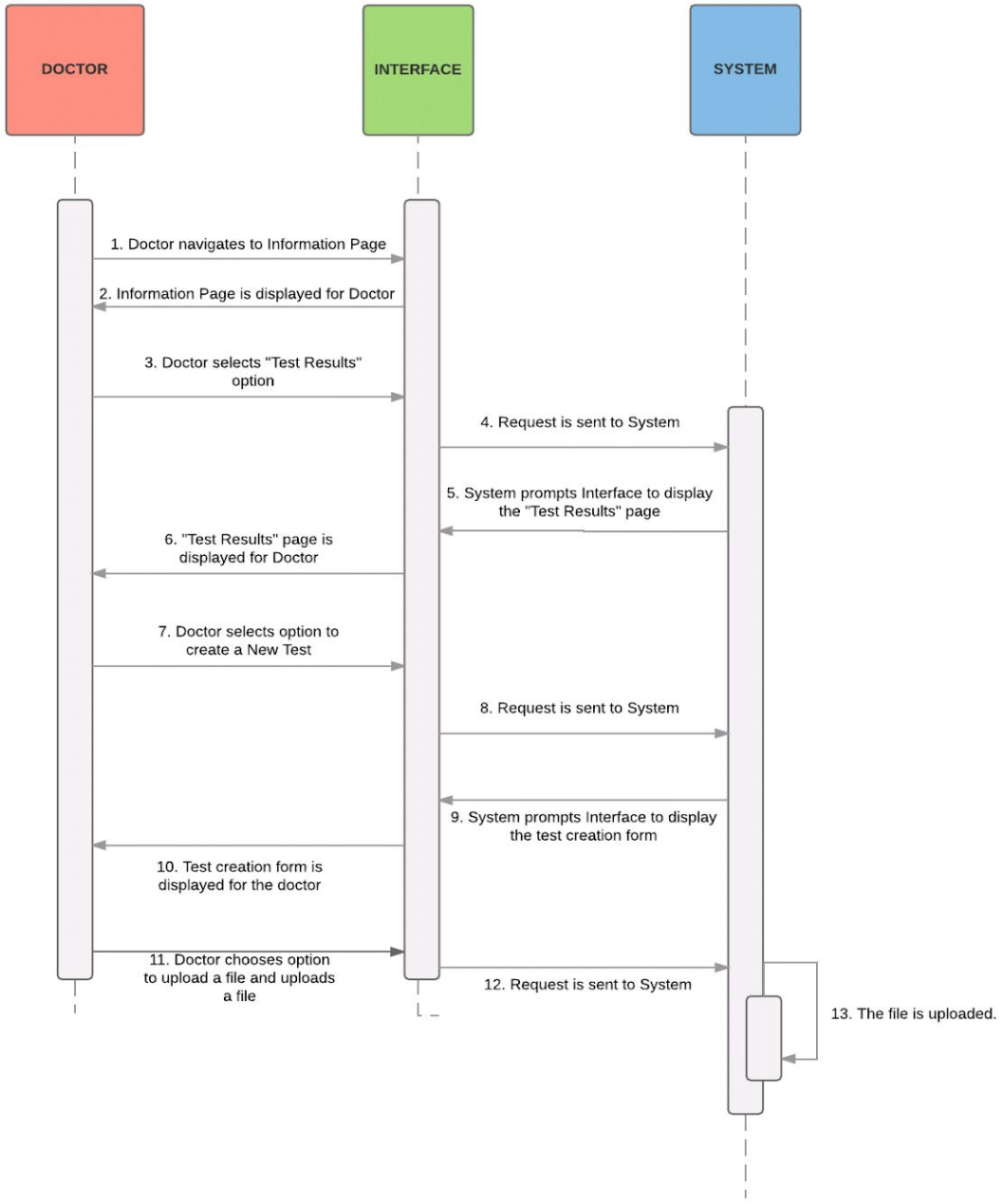
UC 16: PATIENT TRANSFER

Michael Gilmour | December 5th 2016



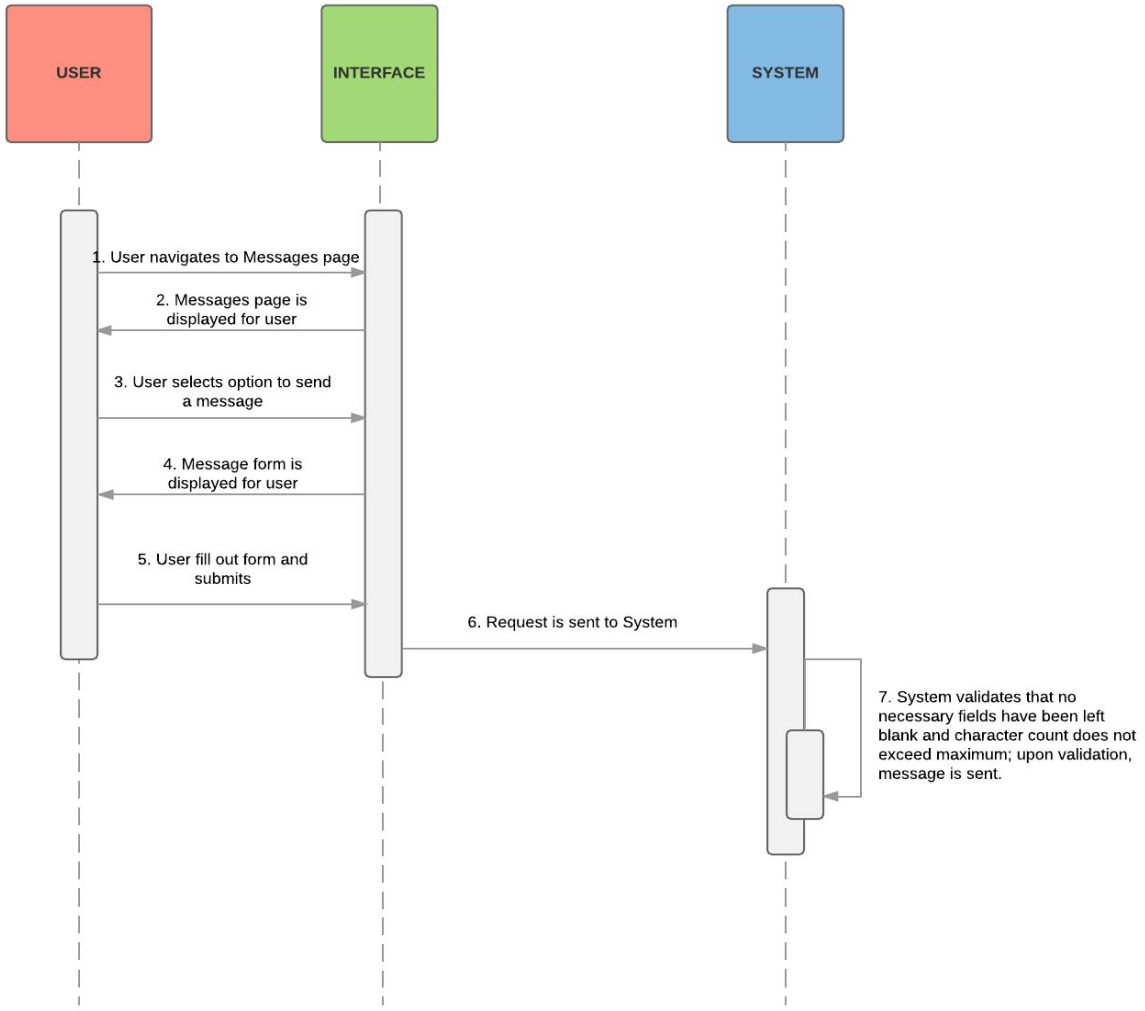
UC 17: UPLOAD PATIENT INFORMATION

Michael Gilmour | December 6, 2016



UC 18: SEND PRIVATE MESSAGE

Michael Gilmour | December 5th, 2016



Design Rationale

- Outside of bugs the main issue was the task of giving our HealthNet app some style and visual flair. At the start, we were shooting mainly for functionality, opting to push off formatting and style for a later time. As R1 got closer, we had a mostly working product with formatting that was subpar to say the least. We were using HTML tables for our formatting (Which is frowned upon by developers). We dedicated a whole day at the end of the R1 Development cycle to transfer over from a table formatting system to using CSS to format our pages. Moving forward, for R2 we will have freed up time by getting our style down to work on core functionality for our 2nd release. The bulk of our 2nd release will be the addition of our additional user type functionality (Doctors, Nurses, Administrators)
- We made our models more efficient and clean by removing any unnecessary models that we had in R1. In R1 we had several models that had overlapping functionality, and going into R2 we both recognized and removed this problem. Doing this made our models file look cleaner, it was easier to read, and we were able to do more with the models we had.
- We decided to use no third party libraries for our product and created every part of our product from scratch. We did this for several reasons: To have more control over the product, to have a better understanding of our product so that we can kill bugs faster, and to have a better learning experience when making our product
- We decided not to use Django's provided models for users and other things of the like. We did this because we wanted to be able to better control our own models and follow our own plan as opposed to changing our plan to fit an already existing model.
- Coming out of R2 Beta we made a decision to pursue a 3rd party calendar called FullCalendar.io. We were disappointed with the complications that arose from our R1 Calendar that made it difficult to add additional functionality. However, when attempting to implement this FullCalendar, we found that integrating a fully comprehensive calendar library to our existing code would be much more difficult than we imagined. Ultimately, despite the flashy opportunity that this 3rd party product presented, we decided to stick with our own JavaScript based calendar and make it work. By throwing a lot of time at the calendar from R1, we were able to refactor the code and add additional functionality as mandated by the

requirements. Now, we can say our calendar is completely original and it meets all the stated requirements. The strengths of this final design is that we know the source code inside-out and can easily modify and add functionality. The deficiency is that the calendar is not as heavily styled as a full-fledged 3rd party calendar. We believe we made the right choice choosing ease of accessibility over style.