Homework 1

Mohammed Algadhib
algadhim@oregonstate.edu
Johnny Chan
chanjoh@oregonstate.edu

Riley Kraft
kraftme@oregonstate.edu

Diego Saer
saerd@oregonstate.edu

ABSTRACT by Stephanie Ayala

Providing quality healthcare for the elderly can sometimes be tricky. One problem is polytherapy leading to the possibility of overmedicating patients and adverse drug interactions. It can be difficult for clients to remember health concerns between appointments and often important topics can be missed. Furthermore, there can be a lack of communication with doctors and miscommunications can arise. Many people with health concerns see multiple doctors for different conditions. During an appointment with one doctor, important information can be overlooked relating to symptoms or medications that other doctors prescribed. This can lead to unnecessary prescription of new medications that interact badly with what the patient is already on, and important symptoms of side effects being missed. As the baby boomers age, this problem will only become larger.

KEYWORDS

user; patient; caretaker; survey; report; notification, drug-drug interaction; side effect; symptom

1. FUNCTIONAL REQUIREMENTS

1.1 DEFINITION

The role of the application is to make it easier for elderly patients and their caretakers to manage medications, avoid dangerous reactions, and mitigate the non-reporting of potentially vital symptoms to their doctors.

The application will feature the functionality to: take a picture of a prescription label and record the details of the medication, as well as give users the flexibility to manually record over-the-counter medications and supplements; keep track of doctor's appointments and medication doses via a calendar; take a guided daily healthy survey to ensure the patient's medications are working as intended; easily provide a written report for patients to take to their doctors' appointments.

Since elderly patients are widely know to take more medications than the average person, and thus are at more risk to adverse effects, this application will also feature an alert system. The alert system is intended to help users be reminded about medication doses and doctor's appointments, and to avoid potentially dangerous reactions to medications.

Patients will also have the option to add one caretaker to co-manage their patient profile. Caretakers receive all notifications that the patient does, but are limited by account permissions given to them by their patient.

The ultimate role of these features is to aid in the reduction of the annual statistics on elderly patients taken to the hospital for adverse side effects of their medications.

1.2 SPECIFICATION

The application is intended for primary use on a mobile device. However, it will also be accessible on a desktop via a web application should a user want to take more control manually managing a patient account.

Before any of the other features of the application can be utilized, a user must record a medication. For ease of use, users take a photo of their prescription label by which important data is extracted and filled into a new prescription form. The user has the ability to edit and/or add additional information, like the prescribing doctor's contact information and medication start date and time, before saving the medication record. For over-the-counter medications and supplements the user must manually enter all pertinent information as the photo capture feature is only compatible with prescription labels.

After a new medication is recorded in the database, the application runs a query with all of the patient's current medications through a reputable open source medication database, much like the DrugBank DB API. If the query returns any new side effects not already in the patient's profile, then they will be added to the database. Similarly, if the query returns any known drug-drug interactions among the patient's medications, then they are also entered into the database and a notification is created and outputted to the patient and, if applicable, their caretaker.

Besides recording medications, the next most important feature of the application is the daily health survey. For ease of use, the user is prompted at the same time everyday, specified by the user, to take the survey. The survey features only a few important questions that required the user to only respond via a 0 to 10 scale. After the user answers the brief questions they have the option to report any pain or discomfort they are feeling in more detail. The application provides a human body diagram, separated into sections, for the user to click on and enter their symptom(s).

Survey data is stored in a database at the end of the survey and available to be queried and printed and/or emailed in a report at the user's will, particularly when they have an upcoming doctor's appointment. After each survey, the patient and, if applicable, their caretaker are notified of any potential problems regarding their medications.

As is common on mobile devices, pop-up notifications can be "swiped" across the screen to be removed. However, this application is calibrated to issue alerts of low, moderate, and severe natures. We don't want a notification of a severe nature to be "swiped" away to be forgotten. Thus, notifications are held in a repository, much like an email inbox, where they are color coded by severity and cannot be deleted. We also don't want the notifications to pile up, thus any notifications past more than 30 days are automatically archived.

Users will also have a calendar by which they can record their medication schedules and doctor's appointments. Medication schedules are automatically generated by the dosage information for their medications, which can be manually altered via the web application if needed. Doctor's appointments must be manually entered like in any calendar app (mobile or web app), but can draw from a list of the patient's known doctor's already recorded in the application.

Before each appointment, the user is prompted by a notification to print a report, which is automatically generated by the software application 24 hours before the appointment. Most doctors' offices ask patients to fill out wellness forms covering their well being over the past two weeks so the doctor has the most relevant information. The software report queries data from the most recent 30 days in order to ensure a patient does not forget to tell the doctor any potentially pertinent information during their appointment.

2. NON-FUNCTIONAL REQUIREMENTS

Reliability: The system should be very reliable. The reliability should be as close to 100% as possible since lives could be at stake. For example, if a patient needs to see a doctor for major surgery and cannot print out a report then there is a chance for drug interactions.

Efficiency: The efficiency of the system does not have to be very high. It should be about average, similar to what users experience with other applications.

Integrity: The integrity of the system must be very high. People will be using the system to manage their health and a faulty interactions warning could be dangerous.

Usability: It should be very easy to use since the elderly and their caretakers are going to be using the software. In this regard, the interfaces will be simplistic (no frills), with large, bright action buttons, and large readable text.

Maintainability: Maintainability should be about average for a similar system. Though needed updates to bugs in the system should be accounted for, changing bits of code is not an important selling point of the application. Too many changes could make elderly patients frustrated with learning new features and less likely to continue to use the system.

Testability: Since reliability and integrity need to be high, the system must be very testable.

Flexibility: Average to low flexibility for a similar application. The application must be simple for the elderly and their caretakers to understand and having many features does not contribute much to the main functionality. Users will have the option to manage their accounts with more flexibility via the web interface, however this is not a priority of the system.

Portability: The system needs to be reasonably portable. It needs to work on PCs, Macs, tablets, and smartphones in order to accommodate the most reasonable technologies that elderly patients may be comfortable using.

Reusability: There is not much need for reusability since the main intent is not to spawn additional apps in the future.

Interoperability: The interoperability needs to be reasonably high as the program includes a critical API that must work with as little problems as possible. The API is required to return known side effects and drug interactions that are potentially dangerous to a patient's life. Without seamless interoperability, the reliability of the system deteriorates.

3. USE CASES

As approved by our customer, though there are several use cases for this software application, the following three were determined to be of the highest priority.

3.1 RECORDING MEDICATIONS

Name: Record a New Prescription Medication

Actor: Elderly Patient or Caretaker

Preconditions:

1. User has a mobile device with a camera.

2. User has downloaded the software application on their mobile device.

3. User has set up their user account and is logged in.

4. Uer has the prescription container and physical medication in front of them.

5. If user is a caretaker, they have been invited, through the software, by the elderly patient to help manage the patient account.

6. If user is a caretaker, they have been given permissions by the patient account to record new medications into the patient's profile.

Postconditions:

1. User successfully recorded the new prescription into the patient's database.

2. Medication has been checked for side effects and interactions with the patient's other medications.

3. Software calendar is populated with the medication dosage schedule.

4. User can identify medication by referencing a stored photograph.

Flow of Events:

1. User taps on New Prescription icon

2. App opens to camera image capture

3. If prescription label is on a small bottle, user taps on panorama camera feature icon

4. User focuses camera over prescription label

5. User taps mobile screen to capture image of prescription label

6. If image cannot be read by the software, screen displays a brief error message (~3 seconds) and returns to event 2.

7. Application translates the medication name, prescription number, doctor's name, expiration date, number of remaining refills, dosage amount and dosage frequency, and records the data in the patient's profile database.

8. App screen changes to a form where the user can make manual changes to the prescription details.

9. User taps on form icon for the medication's profile picture.

10. App opens to camera image capture.

11. User removes a physical sample of their medication from their prescription container.

12. User focuses camera over physical medication.

13. User taps mobile screen to capture image of physical medication.

14. App returns to medication form.

15. User taps on Start icon under medication profile picture.

16. User taps on current date which populates a calendar, by which the user can select an alternative dosage start date, then taps the OK icon.

17. User taps on current time which populates a clock, by which the user can select an alternative dosage start time, then taps the OK icon.

18. User has the option to manual edit the data generated from the prescription label image by tapping on the respective text field and typing in their edits. As well, the user can type in an email address and phone number by which to reach the prescribing doctor.

19. User scrolls to the end of the form and taps the SAVE icon.

20. Screen displays a brief message for 3 seconds indicating their new medication was saved successfully.

21. If a new record could not be created, screen displays a brief error message for 3 seconds and returns to event 18.

22. In the background, app runs all medication records in the patient's database through a known "drug interactions" database from a reputable online source.

23. Any returned interactions generate new notifications that are recorded in a notifications table in the patient’s profile database.

24. A notification block will populate on the user’s mobile screen and, if applicable, on their companion user’s mobile screen.

25. User can swipe screen notifications to the right in order to remove them.

26. App returns to the main menu screen.

3.2 HEALTH SURVEY

Name: Submit the daily health survey

Actor: Elderly Patient or Caretaker

Preconditions:

1. The user has a mobile device or desktop.

2. The software is downloaded to the user’s device.

3. The user has an active account.

4. The user is logged in to the application.

5. The user has received a notification to take their daily survey.

6. The daily report is not submitted yet.

7. If the user is a caretaker, they have been invited to help manage the account by the patient.

8. If the user is a caretaker, they have been given permission to fill the daily health survey.

Postconditions:

1. The survey results are stored in the database.

2. System alerts if there are side effects or medicines are not working.

3. Notification record created if there are any alerts for side effects and not working medicines.

Flow of events:

1. The user taps on “Take Daily Survey” icon.

2. The app opens a new activity to take the survey.

3. The list of medications and their side effects are taken to produce survey questions that detects abnormal side-effects and the effectiveness of each medicine.

4. The survey questions are presented to the user one at a time.

5. The user answers the question by sliding a bar that represents a scale of 0 to 10 from right to left respectively, 0 representing none and 10 is the extreme.

6. After all questions are answered, a human body picture is presented on the screen.

7. The user clicks on a body part to enter the symptoms felt on that part, on that day.

8. A drop down menu is shown to the user to select the symptoms (pain, itch, color, etc).

9. If the symptom is not listed in the drop down menu, the user may select “Other” and type in the symptom manually.

10. The answers for each question are recorded in the database.

11. The symptoms recorded are queried to be compared to the side effects and interactions of the medications taken by the patient.

12. If there are any present side effects, an alert will outputted to the patient and, if applicable, their caretaker.

13. The notification will be color coded based on the severity of the alert, red for high risk, orange for moderate risk, yellow for for low risk, and green for no risk alert.

14. The notification can be swiped to the right by the user to remove it from the screen.

15. The notification and its possible causes will be stored in the database to be reviewed when needed in the future.

16. The event of the daily health survey activity ends and the app returns to the main page.

3.3 REPORTING TO DOCTORS

Name: Generate Health Report for Doctors Appointment

Actors: Elderly Patient or Caretaker

Preconditions:

1. User has an upcoming doctor’s appointment

2. Health report has been generated and recorded in the database.

3. Health report has not been submitted by email or printed.

4. Software is pre-configured on mobile device and PC/Mac and has the latest edition.

5. User account is active on the database.

6. User has logged in and has access to records.

7. User has doctor’s email address attached to the account. If email is not attached with the account, user must print the report and bring it to their appointment.

Postconditions:

1. User has submitted or printed health report in a reasonable time for their appointment.

2. Health report is received by the doctor.

3. Doctor can discuss the report with user at their appointment.

4. Notification is sent to the user after successful submission.

5. Report is available to the user on the mobile app for a predetermined amount of time, typically 30 days.

6. Health report is archived on the database, after 30 days.

Flow of events:

1. User selects the health report icon from the homepage menu.

2. Health report page populates and lists links to all reports generated in the past 30 days from the database.

3. User selects the most recent report, typically no longer than 2 pages.

4. User is presented with the desired report to be submitted.

5. User reviews report data consisting of, chronological symptom events, line charts of severity of symptoms over the past 30 days, and a list of the patient’s medications.

6. User scrolls to the bottom of the report and selects the green SUBMIT icon in bold for readability.

7. Another window pops up with the options to EMAIL or PRINT.

8. If there is no known email for the doctor in which the patient has an appointment, then the EMAIL icon is disabled.

9. If the EMAIL icon is not disabled, and the user selects it, the user’s email application opens with a new message to be composed, populated with the user’s email address, the doctor’s email address, and attached PDF of the report, and a generic message.

9.1 The user has the ability to edit the email.

9.2 User selects the SEND icon in their email application.

9.3 Screen returns to report with EMAIL and PRINT window still active.

9.4 Doctor receives the report and can be seen in the inbox of doctor’s email account. Doctor may receive a different email that the user will print the records and bring it in person.

10. If PRINT icon is selected, the users device defaults to it’s print settings with a PDF of the report in its queue.

10.1 User selects their print settings and then the print icon.

10.2 Report is printed on hard copy.

10.3 If user’s profile contains doctor’s email, the user’s email application opens with a generic message indicating the patient will bring a report in person.

10.4 User edits then sends or exits the email and returns to the software application.

11. User gets a notification on the screen that their report was submitted successfully. Notification will hide after a period of time, about 20 seconds.

12. Health report is archived and stored in the database for future access.

13. User is brought back to the homepage after successful submission.

14. User closes app and user is signed off with no loss in data.

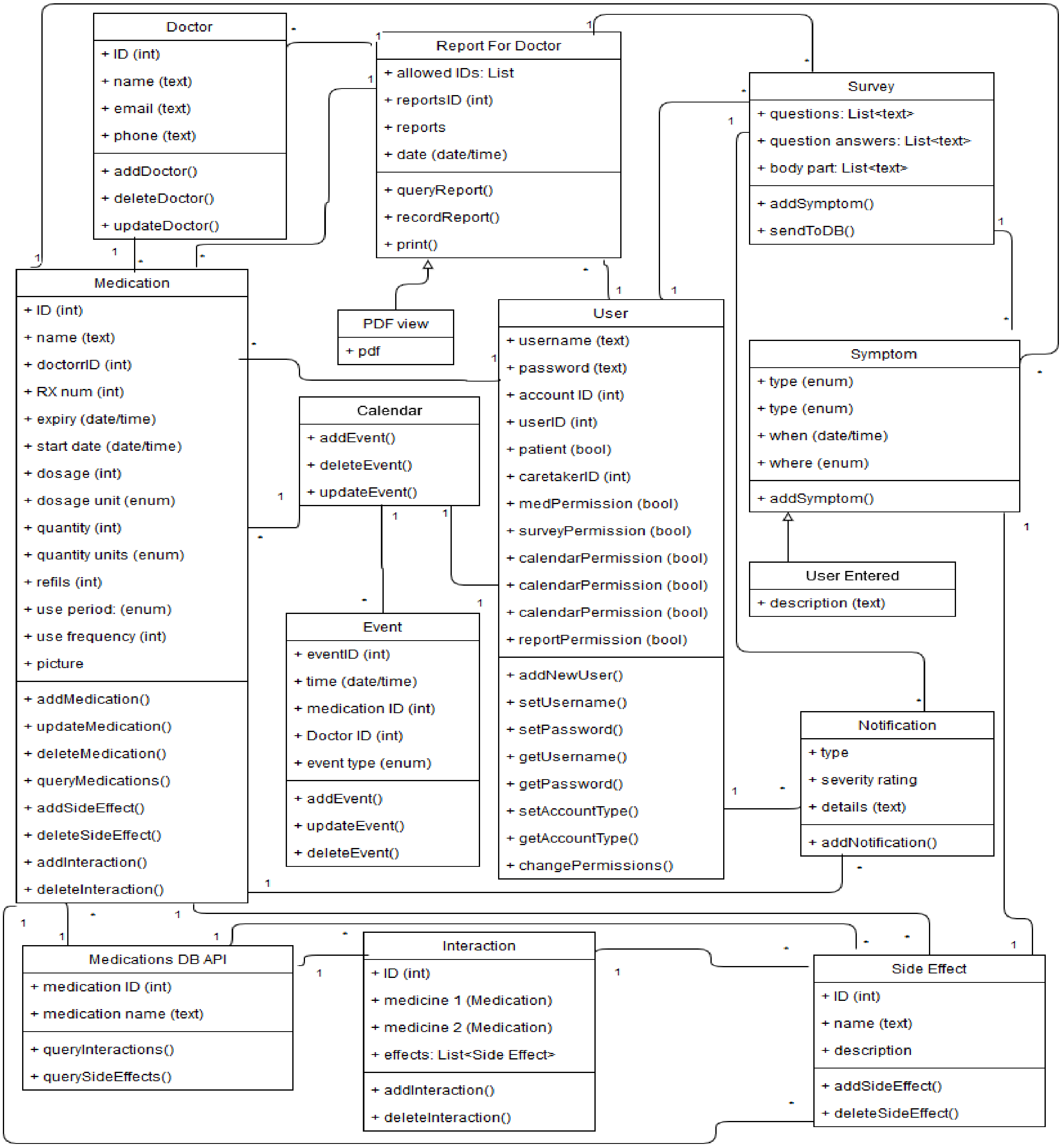
15. User disconnects from the software.

4. UML CLASS DIAGRAM

The diagram below shows all the components (classes) that make up the application. Each box is a class. The boxes are subdivided into 2 or 3 levels arranged vertically. The top level is the class name, below that the class variables, and at the bottom are the class methods. The lines between boxes are the connections between them.

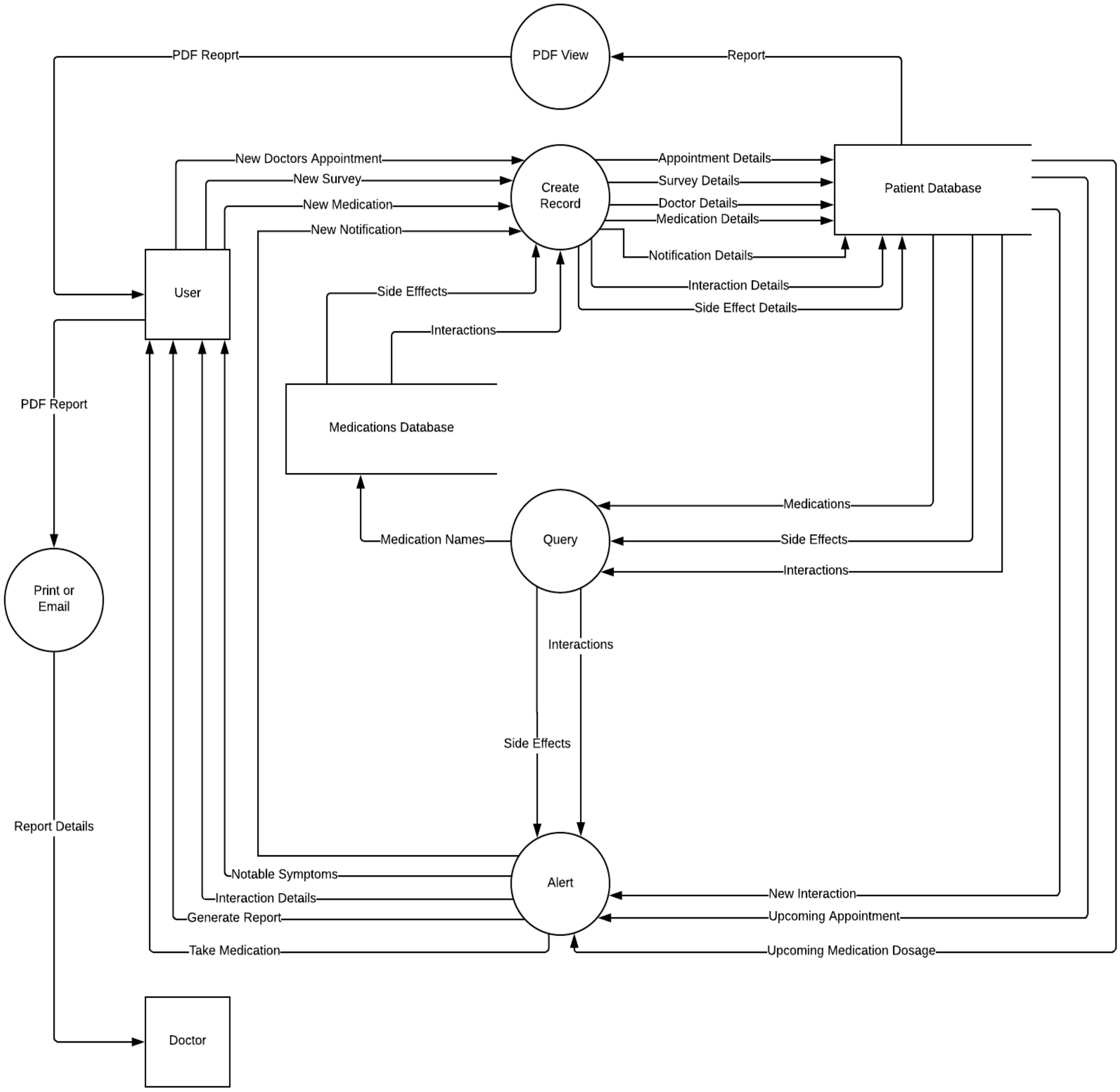
The numbers next to these lines show the cardinality between classes. So if class 1 has a “*” next to it and class 2 has a “1” next to it it means that there are many (potentially infinite) instances of class 1 for every instance of class 2. Finally the arrowheads show dependencies.

The class’ methods indicate the functions used to manage records of class objects in the software database. Each class is a table, each object is a record, and most attributes are columns in their respective tables.



5. DATA-FLOW DIAGRAM

The following diagram depicts the flow of parameters and outputs within the software application system, as a collaboration of the system's use cases. Squares are actors, circles are functions provided by the system, rectangles are data stores, inward arrows are parameters, and outward arrows are outputs. Lucid Charts was used to create the chart, and thus the symbols used reflect the Lucid Charts tutorial for data flow diagrams.



6. USE CASE STATE CHARTS

6.1 RECORDING MEDICATIONS

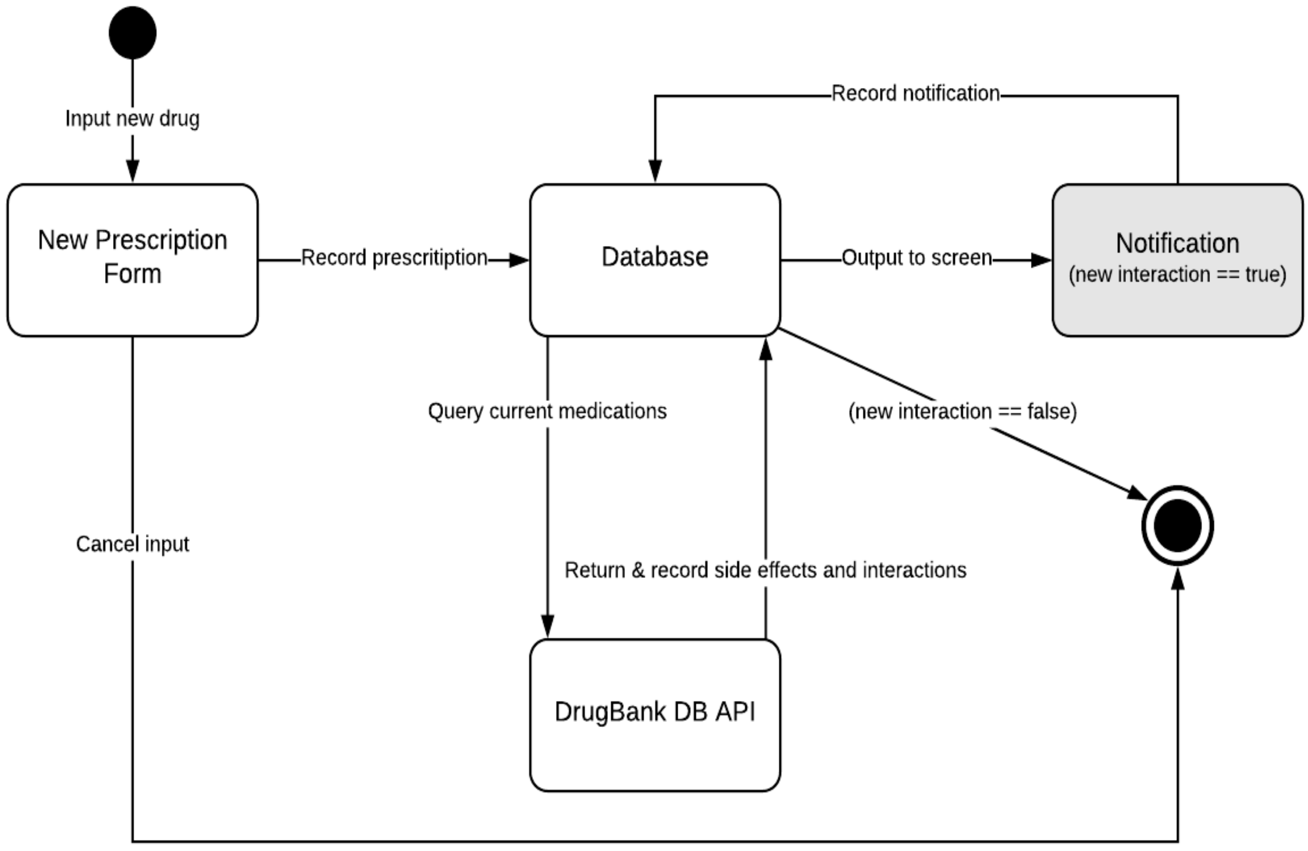
The purpose of this software is to keep track of an elderly patient's medications and reactions in order to help reduce the risk of drug interactions with potentially dangerous side effects. Before the software can monitor a patient's reactions, or report any side effects, the system must first obtain the data for the patient's medications.

In this process, the patient or the patient's caretaker has obtained a new prescription and needs to record it in their application. The user inputs the drug's information into a new prescription form by taking a picture of the prescription label. The application translates the label and generates the relevant data in the form, allowing the user to make manual changes. Once the user has confirmed the correctness of the information, the data is stored in a new medication record in the database.

A new medication record triggers a query which runs all the patient's current medications through a reputable open-source medication database, like the DrugBank API. If the query finds any new side effects or drug-drug interactions among the patient's medications, then the returned records from the query are recorded in the appropriate tables in the database.

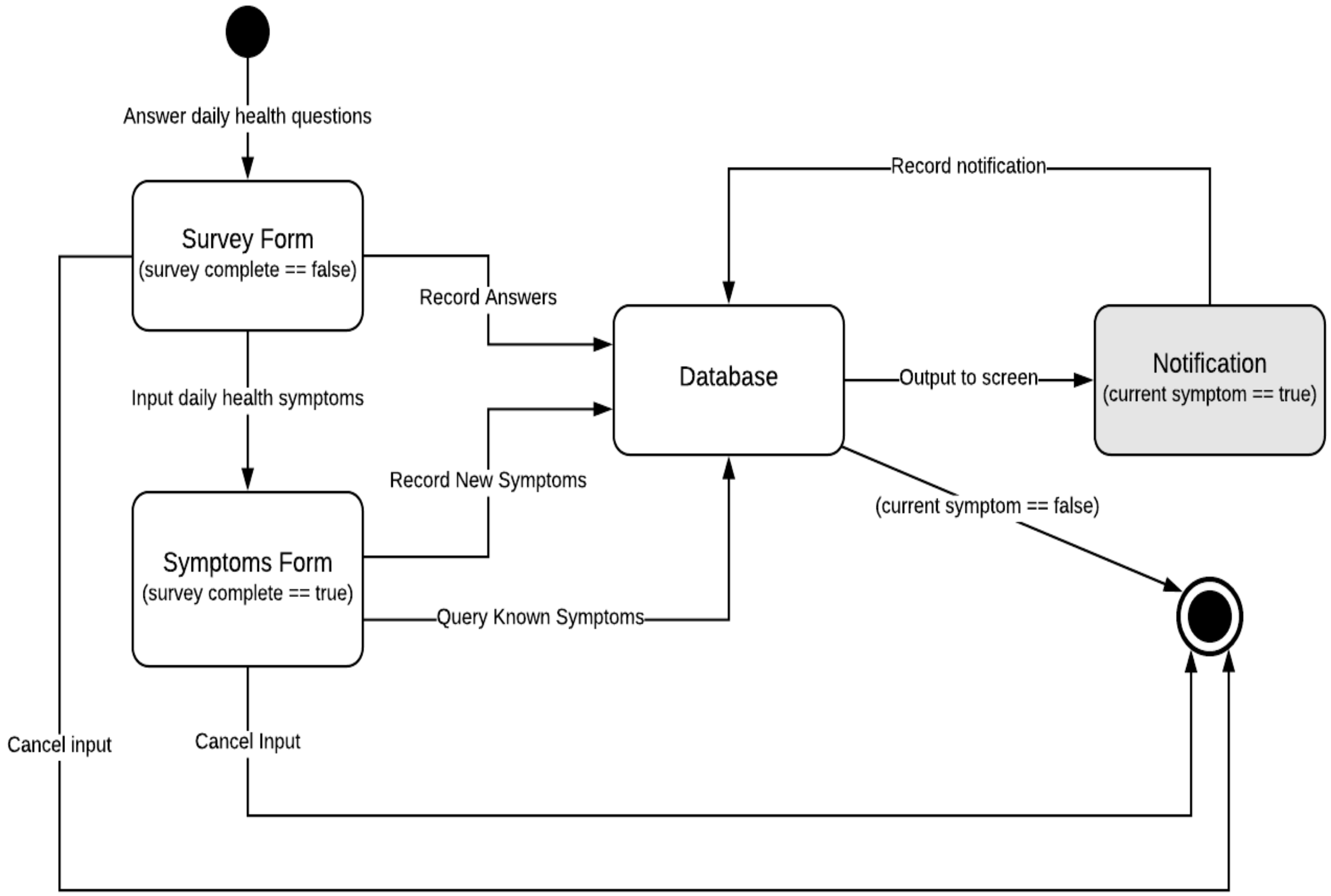
If a new record is recorded in the Interactions table, then a notification is generated and outputted to the patient and their caretaker. The generation of a new notification creates a new record in the Notifications table in the database. Notifications are created for every query record of interactions returned from the API. Each interaction is tagged as "new" until a notification has been issued for it. Therefor, the software stays in a loop of [interaction == new -> output notification -> record notification] until there are no remaining interactions considered "new".

The medication data was confirmed, side effects and interactions have been recorded, and notifications have been issued, thus the process is considered complete.



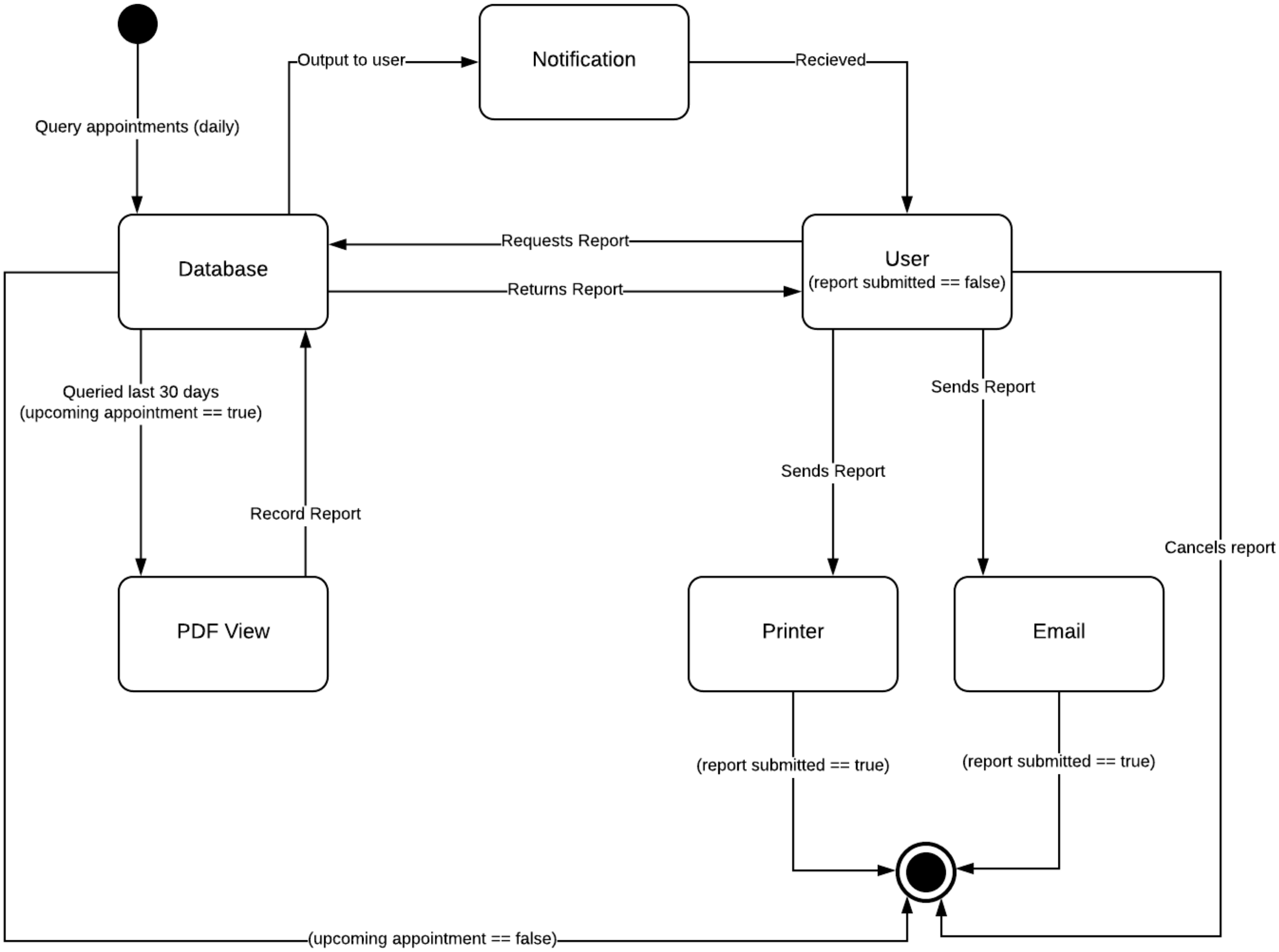
6.2 HEALTH SURVEY

Two of the main functions of the software are to keep track of a patient’s well being and to notify them in case of any symptom they feel is potentially dangerous. This is carried out through a daily health survey. The diagram below shows how data flows through the health survey functionality. The data is created when the user answers the survey questions and inputs their current symptoms, and it is then stored in the database. If the user has input a previously unknown symptom, a new record is created in the Symptoms table with the details of the user’s input. If the user chose a known symptom from the human body image Symptoms Form, then the symptom is queried from the database to return its known severity. Notifications are created for each level of severity of the symptoms entered in the Symptoms Form, and are merged according to said severity. The symptoms entered by the user are tagged as “current” until a notification has been issued, then the tag for “current” is set to false. When there are no more symptoms tagged as “current” the loop ends and the survey exits.


Answer daily health questions
Survey Form
(survey complete == false)
Record Answers
Record notification
Database
Output to screen
Notification
(current symptom == true)
Input daily health symptoms
Record New Symptoms
(current symptom == false)
Symptoms Form
(survey complete == true)
Query Known Symptoms
Cancel input
Cancel Input

6.3 REPORTING TO DOCTORS

In order to ensure the best possible care for the patient a medical report can be printed out and brought to or emailed to their doctor. This report lists the patient's current medicine, chronological symptom events, and statistical charts of the patient's well being survey results. The chart below shows how a report is generated and accessed by the user to give to their doctor. The patient's calendar is queried daily for upcoming doctor's appointments. If there are no appointments within the next 24 hours (default, user may change), then no report is generated and the user is not notified. If the query does return any appointment records, then this triggers a query to analyze and report the patient's data for the past 30 days. The report is created in PDF format and stored for the patient to recover. A notification is then sent to the patient and, if applicable, their caretaker on-screen and/or by email. The user can then utilize request the report from the software to be printed or email to their doctor. After the report has been sent, the process ends. The process can also end by the user cancelling their request for a report.


Notification
Output to user
Recieved
Query appointments (daily)
Database
Requests Report
Returns Report
User
(report submitted == false)
Queried last 30 days
(upcoming appointment == true)
Record Report
Sends Report
Sends Report
Cancels report
PDF View
Printer
Email
(report submitted == true)
(report submitted == true)
(upcoming appointment == false)

7. REFERENCES

[1] DrugBank API.
https://docs.drugbankplus.com/v1/#introduction

[2] Lucid Charts. *What is a Data Flow Diagram*.
https://www.lucidchart.com/pages/data-flow-diagram